Cisco DevNet Series

1. Intro to Software & Programmability
2. Intro to Coding
3. Intent Networks: How to be a Network Engineer in a Programmable Age
4. Fast Lane: Where Code (Apple) Meets Network Infrastructure (Cisco)
5. APIs with Cisco Spark
6. Network Programmability & APIC-EM – Today!
7. Network Programmability with YANG/NETCONF/RESTCONF – March 15

All Series Details can be Found @ http://bit.ly/DevNetSeries

Network Programmability & APIC-EM

Cisco DevNet Webinar Series

Speaker: Matt DeNapoli| DevNet Developer Evangelist

Hostess: Kara Sullivan | Cisco Networking Academy

18 January 2018

Welcome to the 6th session of the Cisco DevNet webinar series

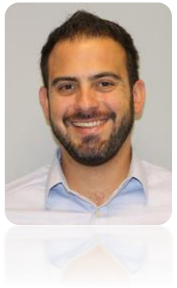
- Use the Q and A panel to ask questions.
- Use the Chat panel to communicate with attendees and panelists.
- A link to a recording of the session will be sent to all registered attendees.
- Please take the feedback survey at the end of the webinar.

Cisco DevNet Series

1. Intro to Software & Programmability
2. Intro to Coding
3. Intent Networks: How to be a Network Engineer in a Programmable Age
4. Fast Lane: Where Code (Apple) Meets Network Infrastructure (Cisco)
5. APIs with Cisco Spark
6. Network Programmability & APIC-EM
7. Network Programmability with YANG/NETCONF/RESTCONF – March 15

All Series Details can be Found @ http://bit.ly/DevNetSeries

Joining You Today:



Matt Denapoli
Developer Evangelist
DevNet, Cisco

Recommended knowledge to follow along today:

- CCNA2
- Basic Programming Skills

Module 03

Network Programmability and APIC-EM

Matthew DeNapoli

DevNet Developer Evangelist

Networking Academy | DEVNET

DevNet › Discover › Learning Tracks › DevNet Beginner › Network Programmability

https://learninglabs.cisco.com/tracks/devnet-beginner

- Networking 101 Basics and Software Defined Networks
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/networking-101-the-basics/step/1
- What is Network Programmability?
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/02-dna-02-what-is-network-prog/step/1
- Controller Basics and APIC-EM
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/05-apic-01-controller-basics-and-apic-em/step/1
- APIC-EM Applications and Use Cases
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/05-apic-02-apic-em-applications-and-use-cases/step/1
- Coding 101 - REST API Basics
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/coding-101-rest-basics-ga/step/1

DevNet › Discover › Learning Tracks › DevNet Beginner › Network Programmability

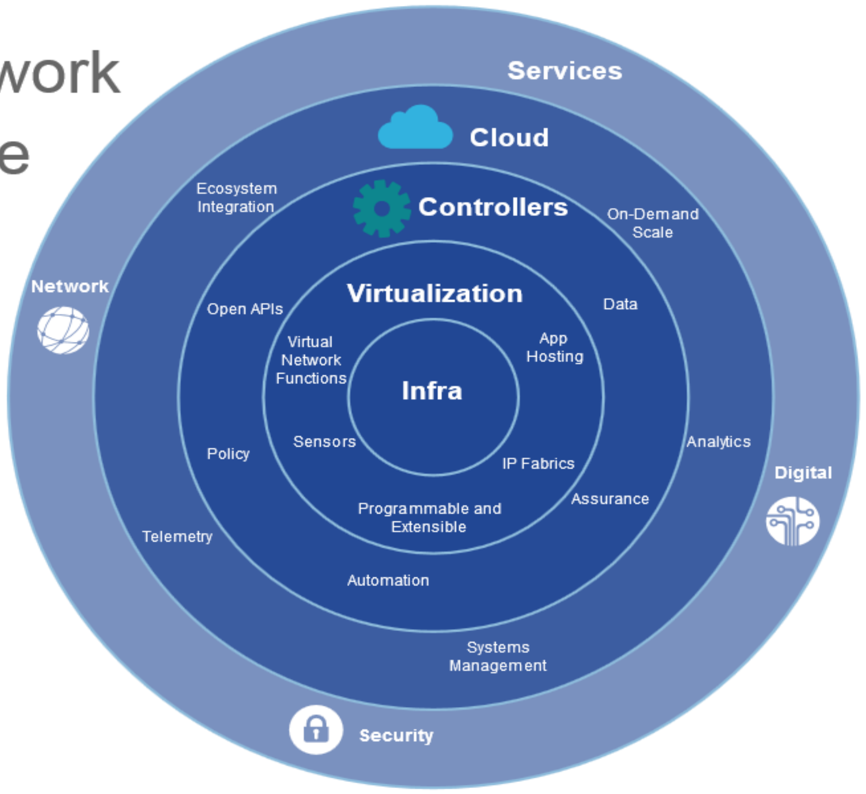
https://learninglabs.cisco.com/tracks/devnet-beginner

- APIC-EM APIs with Python: Part I - The Basics
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/apic-em-1-3-basic/step/1
- APIC-EM APIs with Python: Part II - Path Trace
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/apic-em-1-3-path-trace/step/1
- APIC-EM APIs with Python: Part III - Policy Labs
  https://learninglabs.cisco.com/tracks/devnet-beginner/network-programmability/apic-em-1-3-policy/step/1

Network Programmability, DNA, Controllers

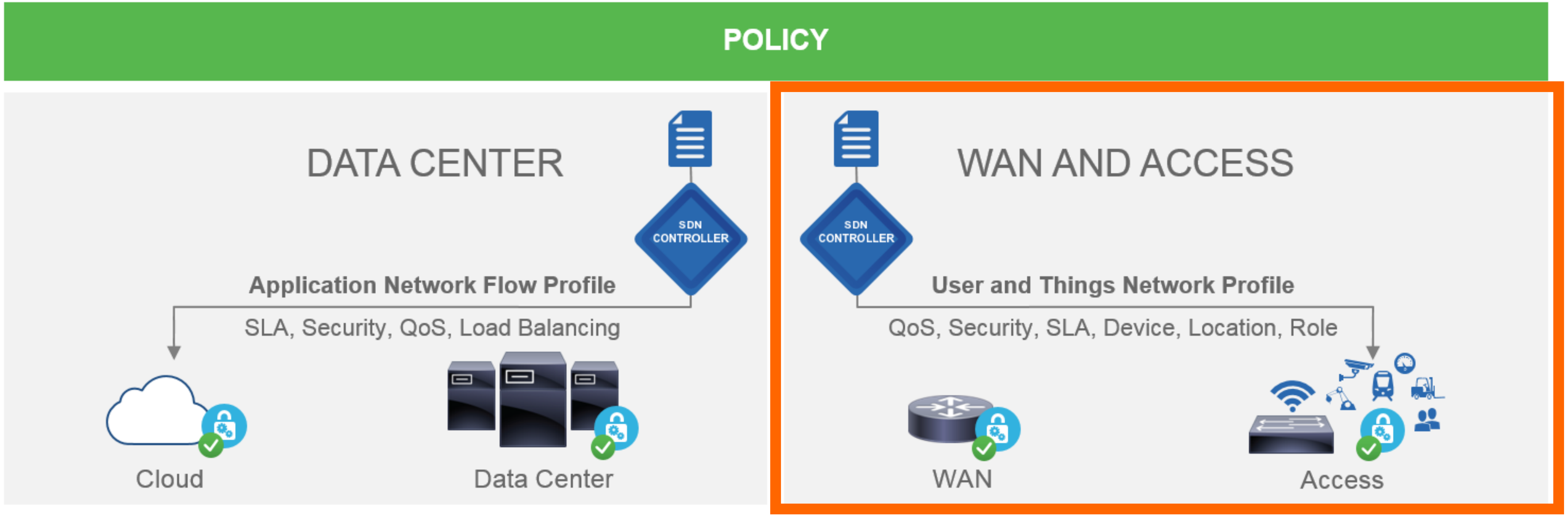
Digital Network Architecture

Digital Network Architecture (DNA) Vision



Open and Software-Driven

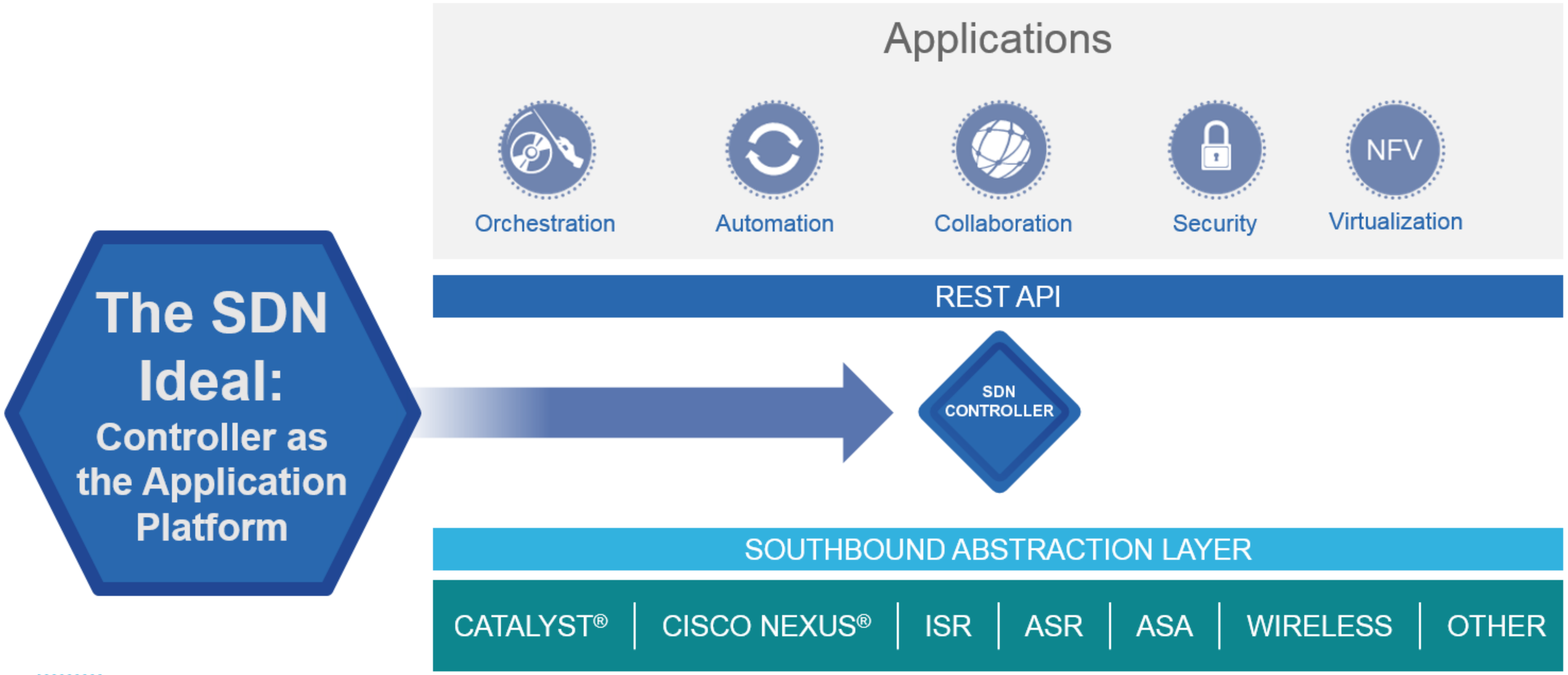
Common Policy Model from Branch to Data Center


POLICY
DATA CENTER
SDN CONTROLLER
Application Network Flow Profile
SLA, Security, QoS, Load Balancing
Cloud
Data Center
WAN AND ACCESS
SDN CONTROLLER
User and Things Network Profile
QoS, Security, SLA, Device, Location, Role
WAN
Access

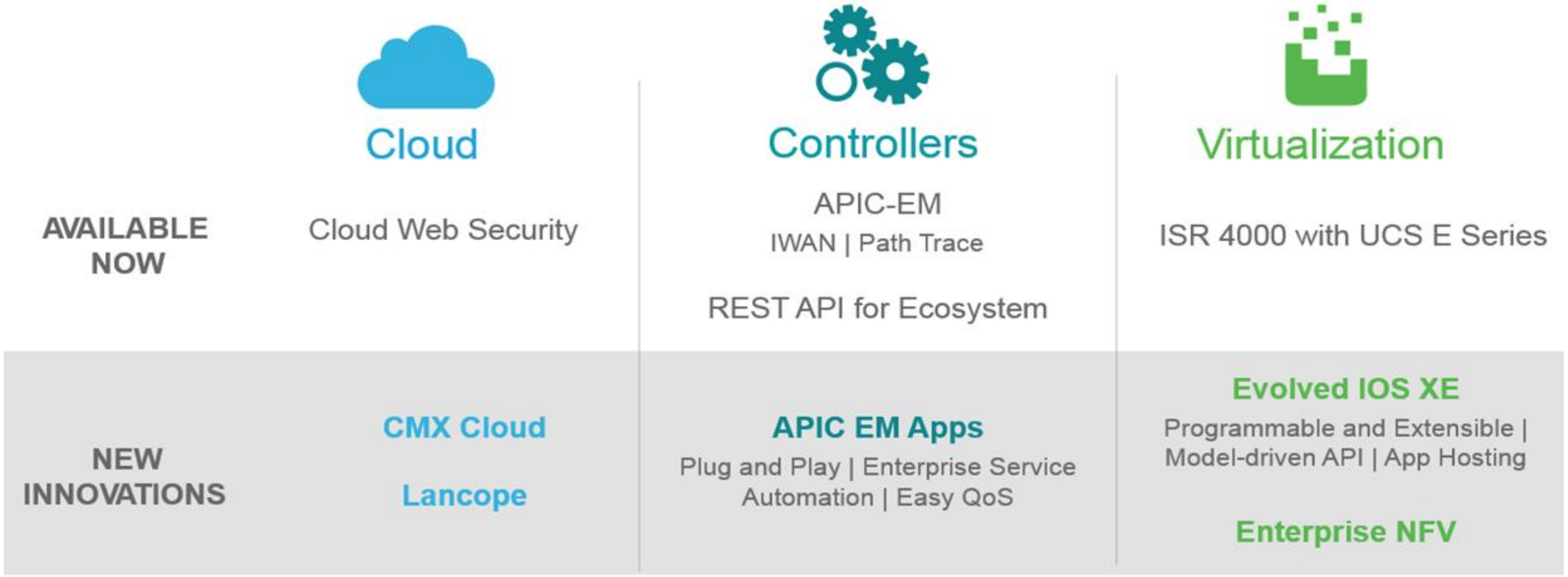
CISCO® ADVANTAGE

BROWNFIELD AND GREENFIELD	END TO END	POLICY FRAMEWORK: FOCUS ON APPLICATION AND USER ENABLEMENT

Network-Wide Abstractions Simplify the Network



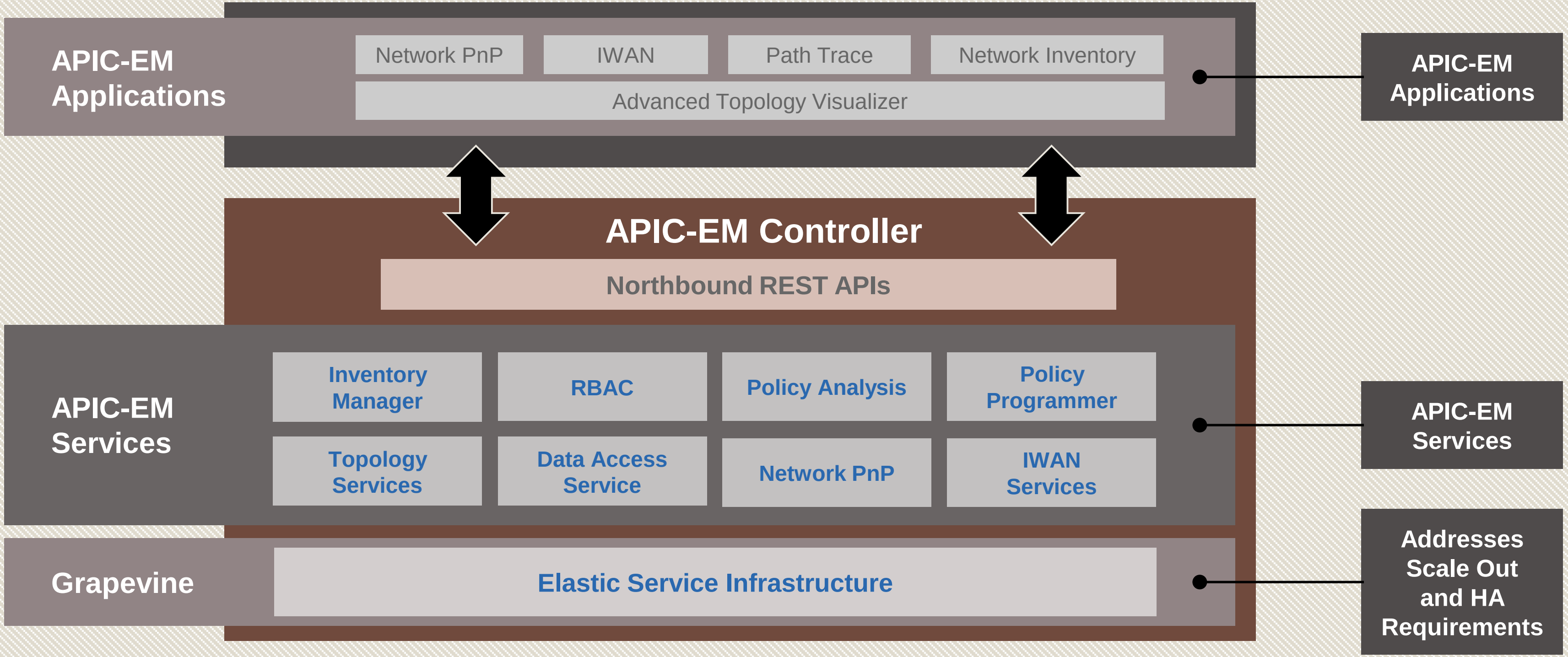
What's New: DNA Innovations



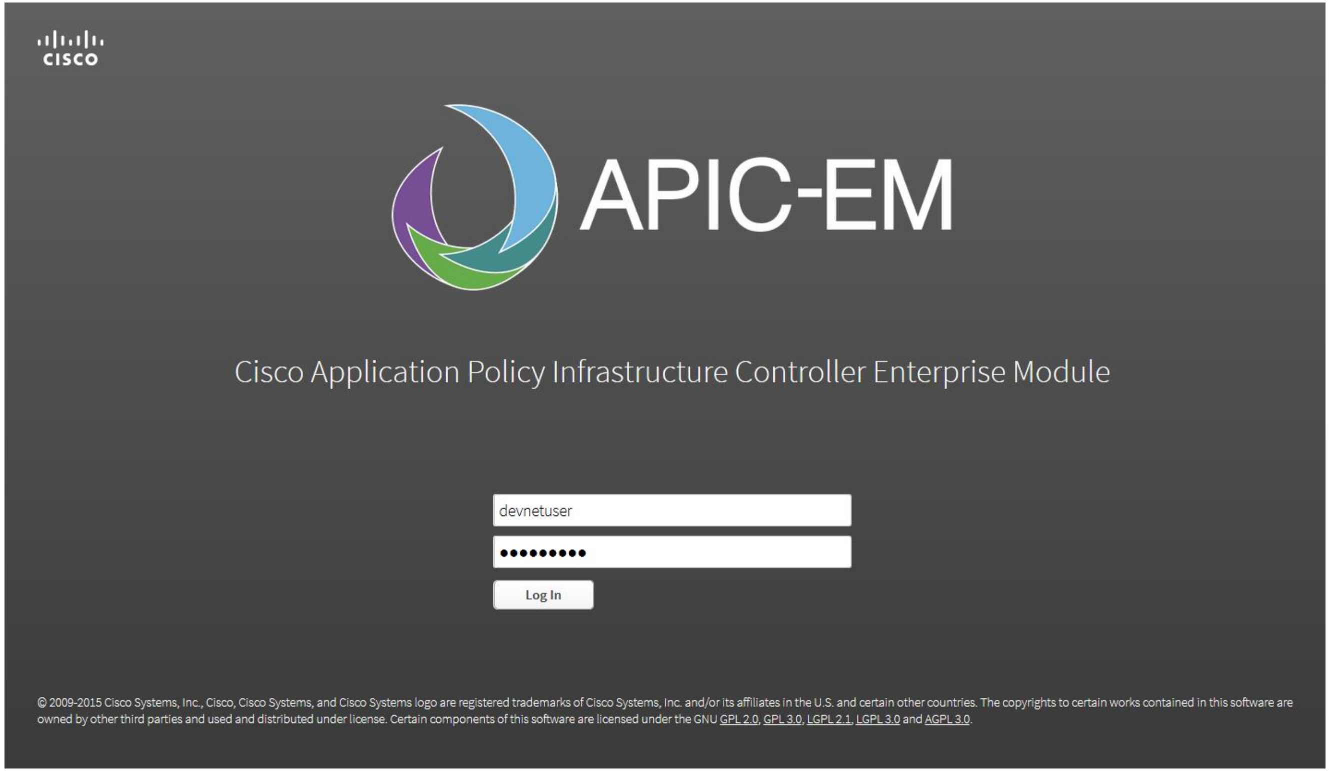
	Cloud	Controllers	Virtualization
AVAILABLE NOW	Cloud Web Security	APIC-EM IWAN | Path Trace REST API for Ecosystem	ISR 4000 with UCS E Series
NEW INNOVATIONS	**CMX Cloud** **Lancope**	**APIC EM Apps** Plug and Play | Enterprise Service Automation | Easy QoS	**Evolved IOS XE** Programmable and Extensible | Model-driven API | App Hosting **Enterprise NFV**

Introduction to APIC-EM

APIC-EM - Platform Architecture

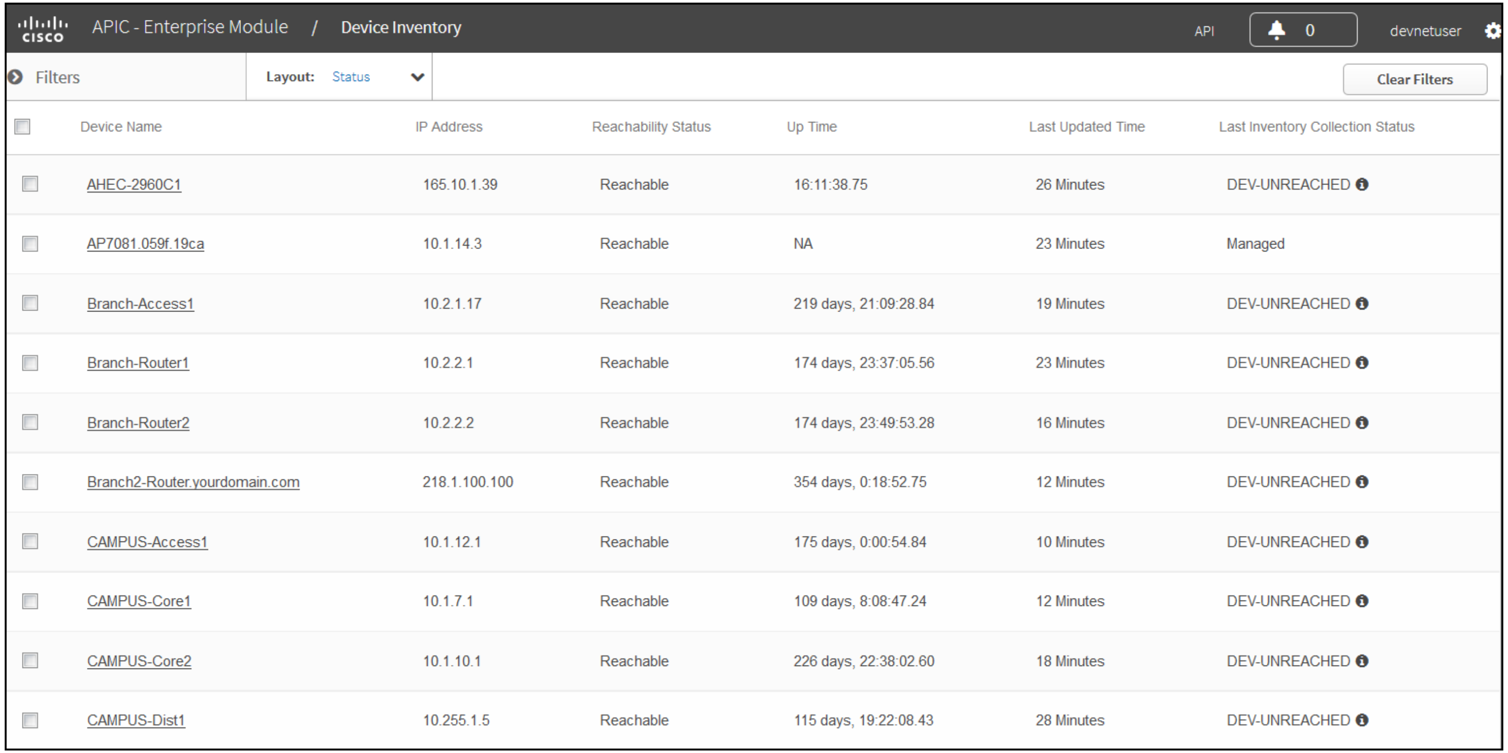


APIC-EM – Log in


APIC-EM
Cisco Application Policy Infrastructure Controller Enterprise Module
devnetuser
●●●●●●●●●●
Log In

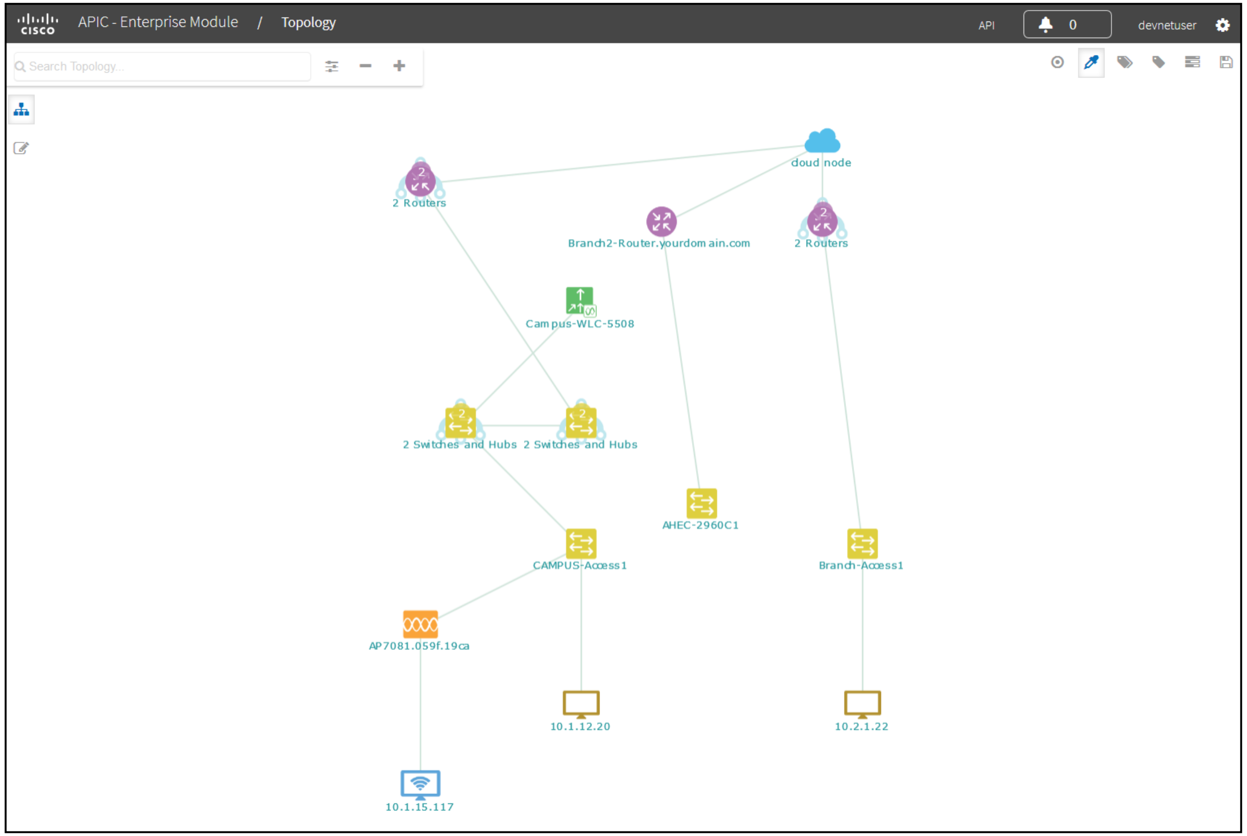
© 2009-2015 Cisco Systems, Inc., Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. The copyrights to certain works contained in this software are owned by other third parties and used and distributed under license. Certain components of this software are licensed under the GNU GPL 2.0, GPL 3.0, LGPL 2.1, LGPL 3.0 and AGPL 3.0.

APIC-EM Device Inventory

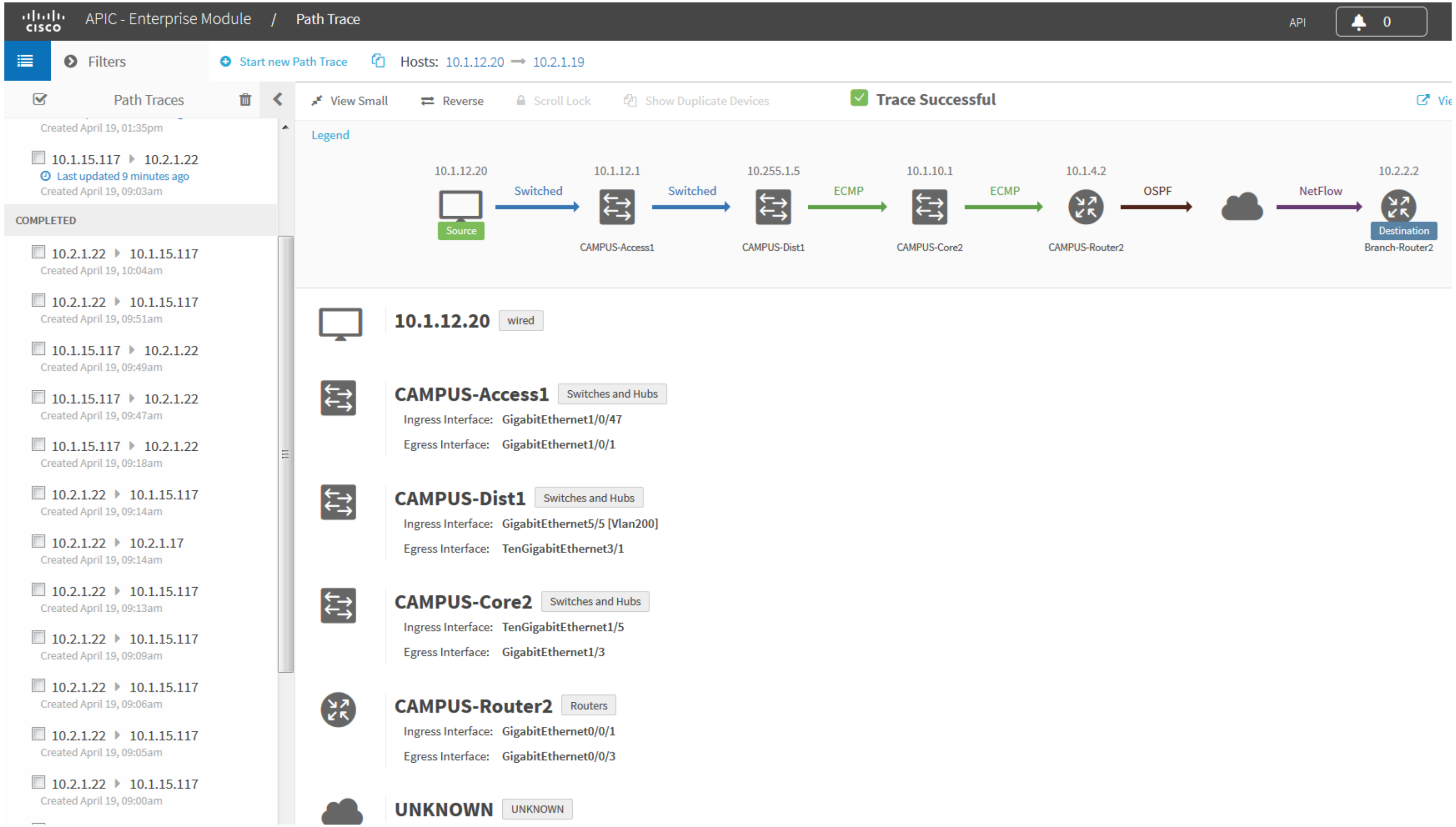


Device Name	IP Address	Reachability Status	Up Time	Last Updated Time	Last Inventory Collection Status
AHEC-2960C1	165.10.1.39	Reachable	16:11:38.75	26 Minutes	DEV-UNREACHED
AP7081.059f.19ca	10.1.14.3	Reachable	NA	23 Minutes	Managed
Branch-Access1	10.2.1.17	Reachable	219 days, 21:09:28.84	19 Minutes	DEV-UNREACHED
Branch-Router1	10.2.2.1	Reachable	174 days, 23:37:05.56	23 Minutes	DEV-UNREACHED
Branch-Router2	10.2.2.2	Reachable	174 days, 23:49:53.28	16 Minutes	DEV-UNREACHED
Branch2-Router.yourdomain.com	218.1.100.100	Reachable	354 days, 0:18:52.75	12 Minutes	DEV-UNREACHED
CAMPUS-Access1	10.1.12.1	Reachable	175 days, 0:00:54.84	10 Minutes	DEV-UNREACHED
CAMPUS-Core1	10.1.7.1	Reachable	109 days, 8:08:47.24	12 Minutes	DEV-UNREACHED
CAMPUS-Core2	10.1.10.1	Reachable	226 days, 22:38:02.60	18 Minutes	DEV-UNREACHED
CAMPUS-Dist1	10.255.1.5	Reachable	115 days, 19:22:08.43	28 Minutes	DEV-UNREACHED

APIC-EM Topology



APIC-EM : Path Trace



APIC - Enterprise Module / Path Trace

API 0

Filters

Start new Path Trace

Hosts: 10.1.12.20 → 10.2.1.19

Path Traces

Created April 19, 01:35pm

10.1.15.117 ▸ 10.2.1.22
Last updated 9 minutes ago
Created April 19, 09:03am

COMPLETED

10.2.1.22 ▸ 10.1.15.117
Created April 19, 10:04am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:51am

10.1.15.117 ▸ 10.2.1.22
Created April 19, 09:49am

10.1.15.117 ▸ 10.2.1.22
Created April 19, 09:47am

10.1.15.117 ▸ 10.2.1.22
Created April 19, 09:18am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:14am

10.2.1.22 ▸ 10.2.1.17
Created April 19, 09:14am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:13am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:09am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:06am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:05am

10.2.1.22 ▸ 10.1.15.117
Created April 19, 09:00am

View Small   Reverse   Scroll Lock   Show Duplicate Devices

✓ Trace Successful

Legend

10.1.12.20 wired

CAMPUS-Access1 Switches and Hubs
Ingress Interface: GigabitEthernet1/0/47
Egress Interface: GigabitEthernet1/0/1

CAMPUS-Dist1 Switches and Hubs
Ingress Interface: GigabitEthernet5/5 [Vlan200]
Egress Interface: TenGigabitEthernet3/1

CAMPUS-Core2 Switches and Hubs
Ingress Interface: TenGigabitEthernet1/5
Egress Interface: GigabitEthernet1/3

CAMPUS-Router2 Routers
Ingress Interface: GigabitEthernet0/0/1
Egress Interface: GigabitEthernet0/0/3

UNKNOWN UNKNOWN

APIC-EM REST APIs

What is so great about REST?

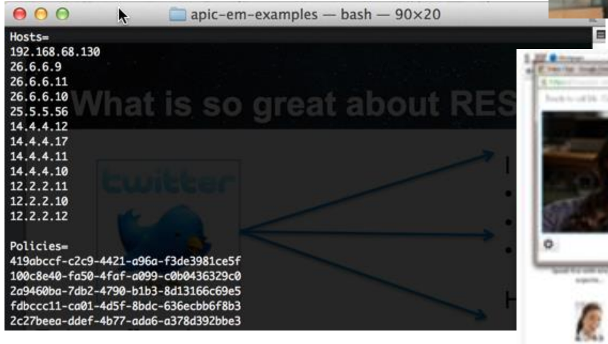
Easy to use:

- In mobile apps
- In console apps
- In web apps

Cisco APIC-EM REST APIs

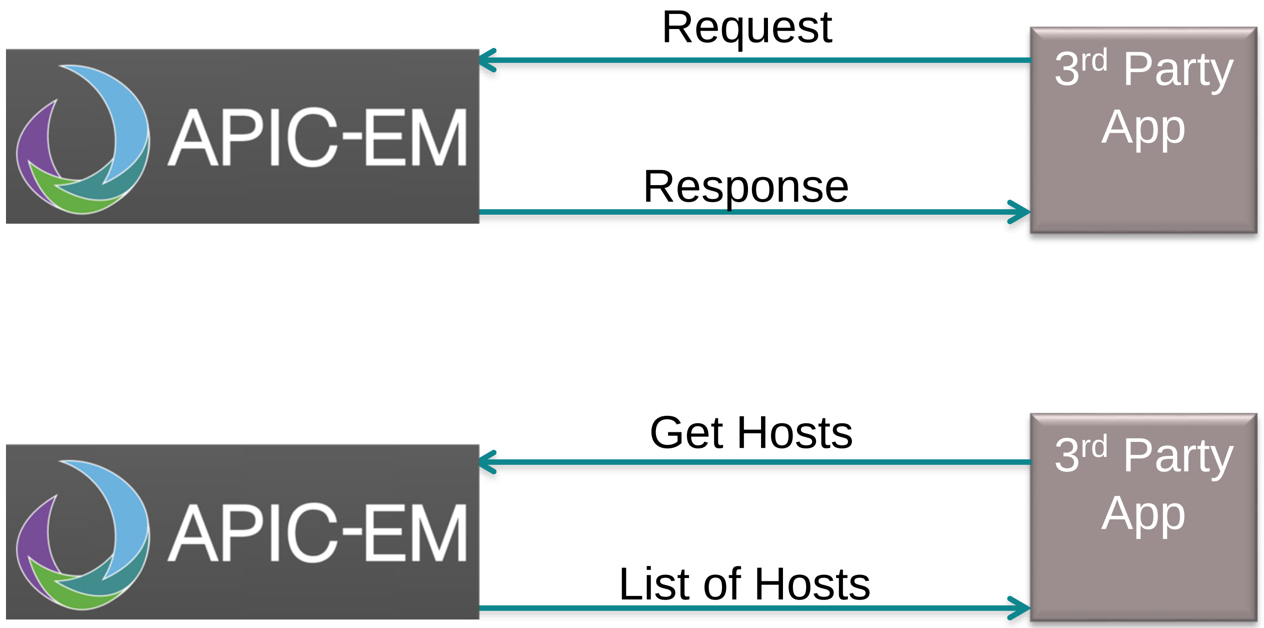
- Hosts
- Devices
- Users
- + more

How does this work?



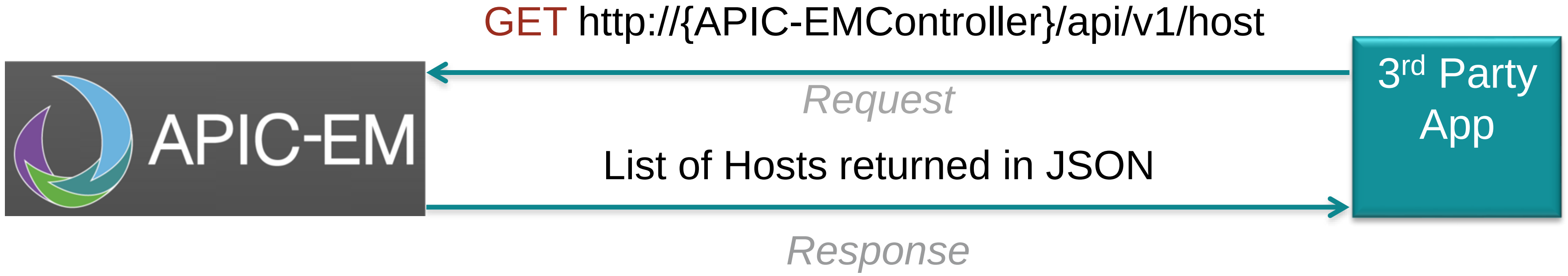
```
Hosts=
192.168.68.130
26.6.6.9
26.6.6.11
26.6.6.10
25.5.5.56
14.4.4.12
14.4.4.17
14.4.4.11
14.4.4.10
12.2.2.11
12.2.2.10
12.2.2.12

Policies=
419abccf-c2c9-4421-a96a-f3de3981ce5f
100c8e40-fa50-4faf-a099-c0b0436329c0
2a9460ba-7db2-4790-b1b3-8d13166c69e5
fdbccc11-ca01-4d5f-8bdc-636ecbb6f8b3
2c27beea-ddef-4b77-ada6-a378d392bbe3
```

How does this work?



APIC-EM Example: Get Host

Application Policy Infrastructure Controller (APIC) Enterprise Module (EM)


GET http://{APIC-EMController}/api/v1/host
APIC-EM
3rd Party App
Request
List of Hosts returned in JSON
Response

Anatomy of a REST Request

Method

- GET, POST, PUT, DELETE

URL

- Example: http://{APIC-EMController}/api/v1/host

Authentication

- Basic HTTP, OAuth, none, Custom

Custom Headers

- HTTP Headers
- Example: Content-Type: application/json

Request Body

- JSON or XML containing data needed to complete request

And what is in the Response?

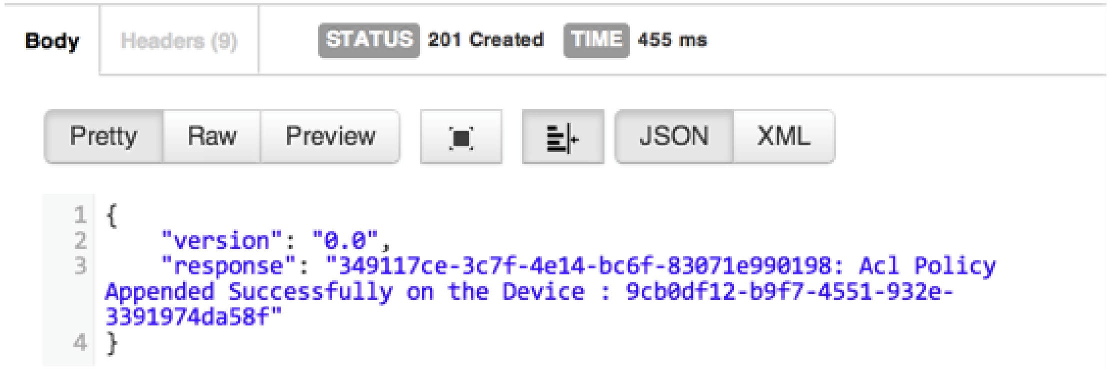
HTTP Status Codes

- http://www.w3.org/Protocols/HTTP/HTRESP.html
- 200 OK
- 201 Created
- 500 Internal Error

Headers

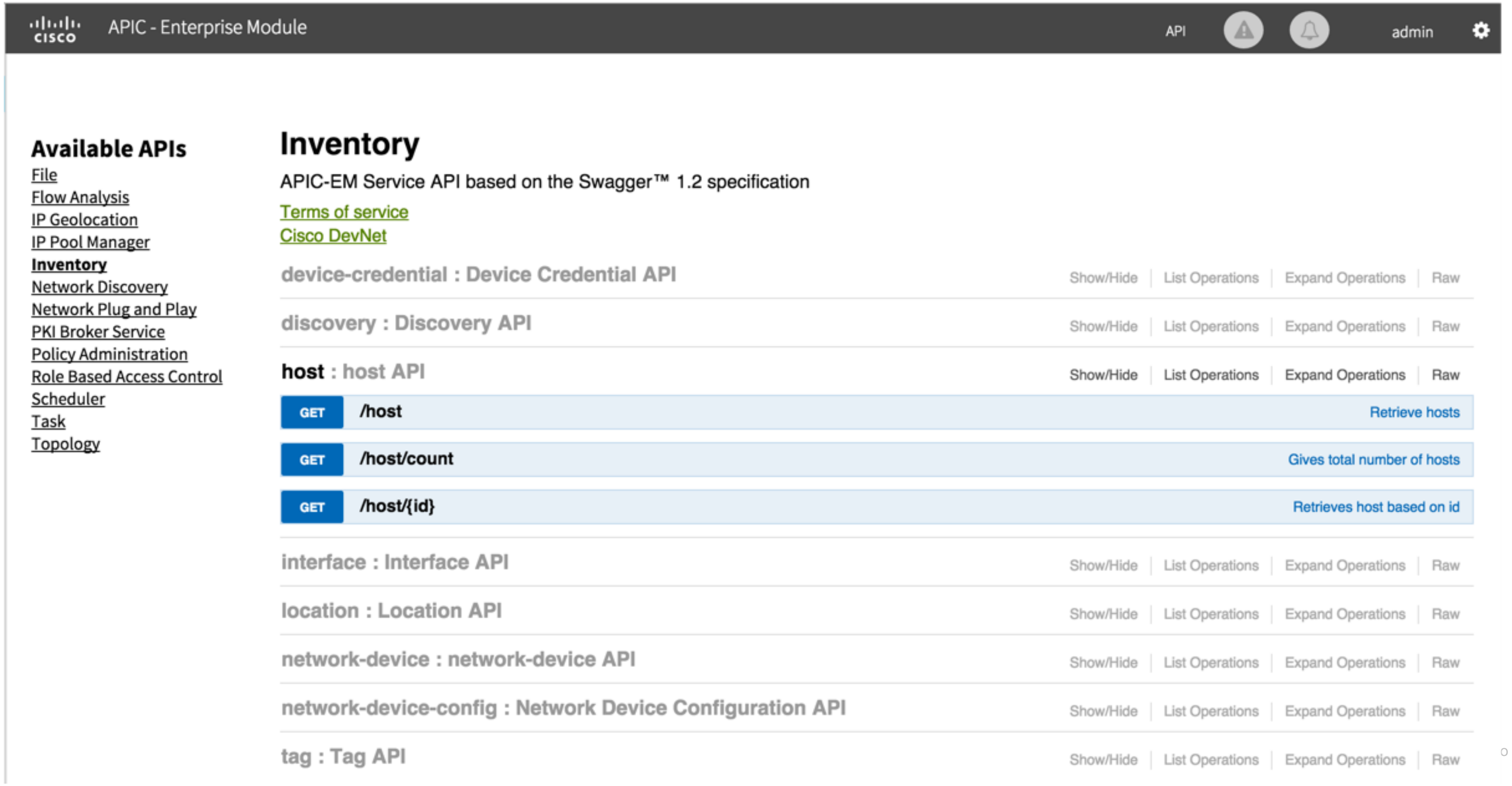
Body

- JSON
- XML



```
{
    "version": "0.0",
    "response": "349117ce-3c7f-4e14-bc6f-83071e990198: Acl Policy
Appended Successfully on the Device : 9cb0df12-b9f7-4551-932e-
3391974da58f"
}
```

Using the API Reference Documentation



APIC - Enterprise Module

API admin

Available APIs

- File
- Flow Analysis
- IP Geolocation
- IP Pool Manager
- **Inventory**
- Network Discovery
- Network Plug and Play
- PKI Broker Service
- Policy Administration
- Role Based Access Control
- Scheduler
- Task
- Topology

Inventory

APIC-EM Service API based on the Swagger™ 1.2 specification

Terms of service
Cisco DevNet

device-credential : Device Credential API — Show/Hide | List Operations | Expand Operations | Raw

discovery : Discovery API — Show/Hide | List Operations | Expand Operations | Raw

host : host API — Show/Hide | List Operations | Expand Operations | Raw

GET	/host	Retrieve hosts
GET	/host/count	Gives total number of hosts
GET	/host/{id}	Retrieves host based on id

interface : Interface API — Show/Hide | List Operations | Expand Operations | Raw

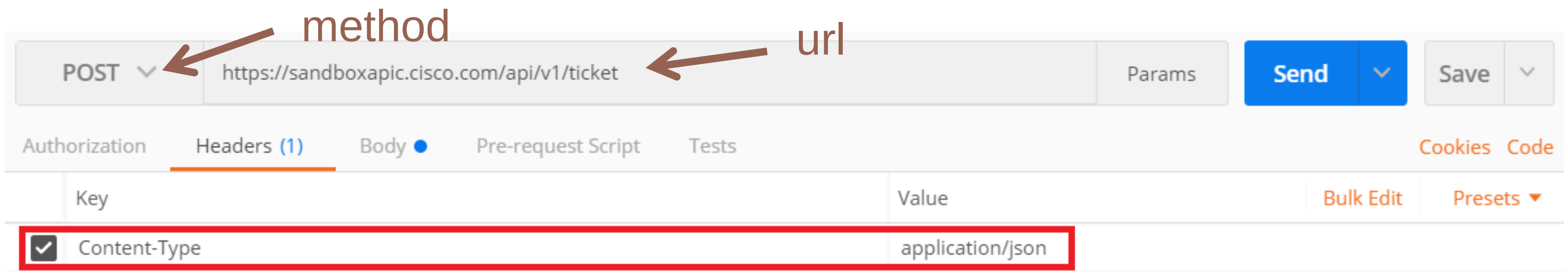
location : Location API — Show/Hide | List Operations | Expand Operations | Raw

network-device : network-device API — Show/Hide | List Operations | Expand Operations | Raw

network-device-config : Network Device Configuration API — Show/Hide | List Operations | Expand Operations | Raw

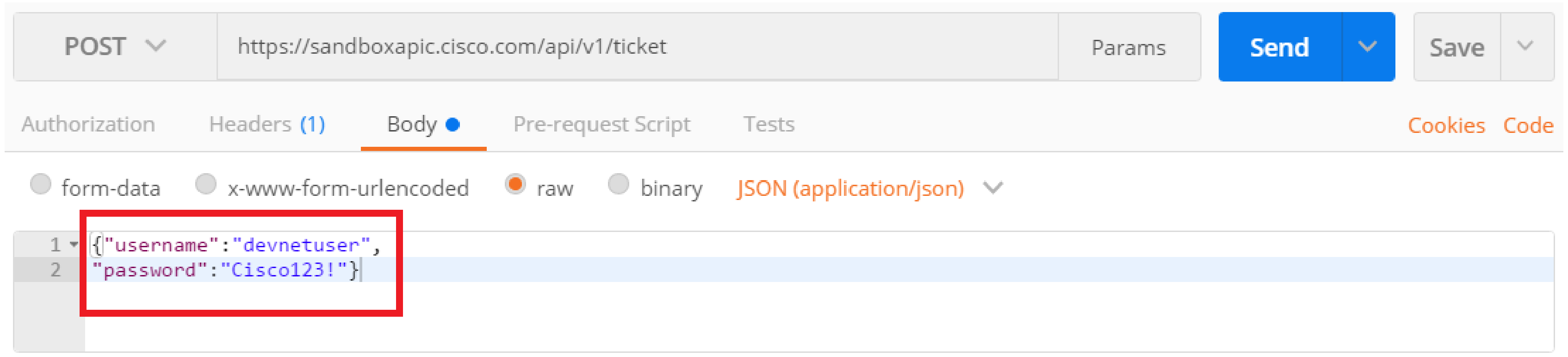
tag : Tag API — Show/Hide | List Operations | Expand Operations | Raw

Using Postman to get the Service Ticket



https://sandboxapic.cisco.com/api/v1/ticket

Key	Value
✓ Content-Type	application/json

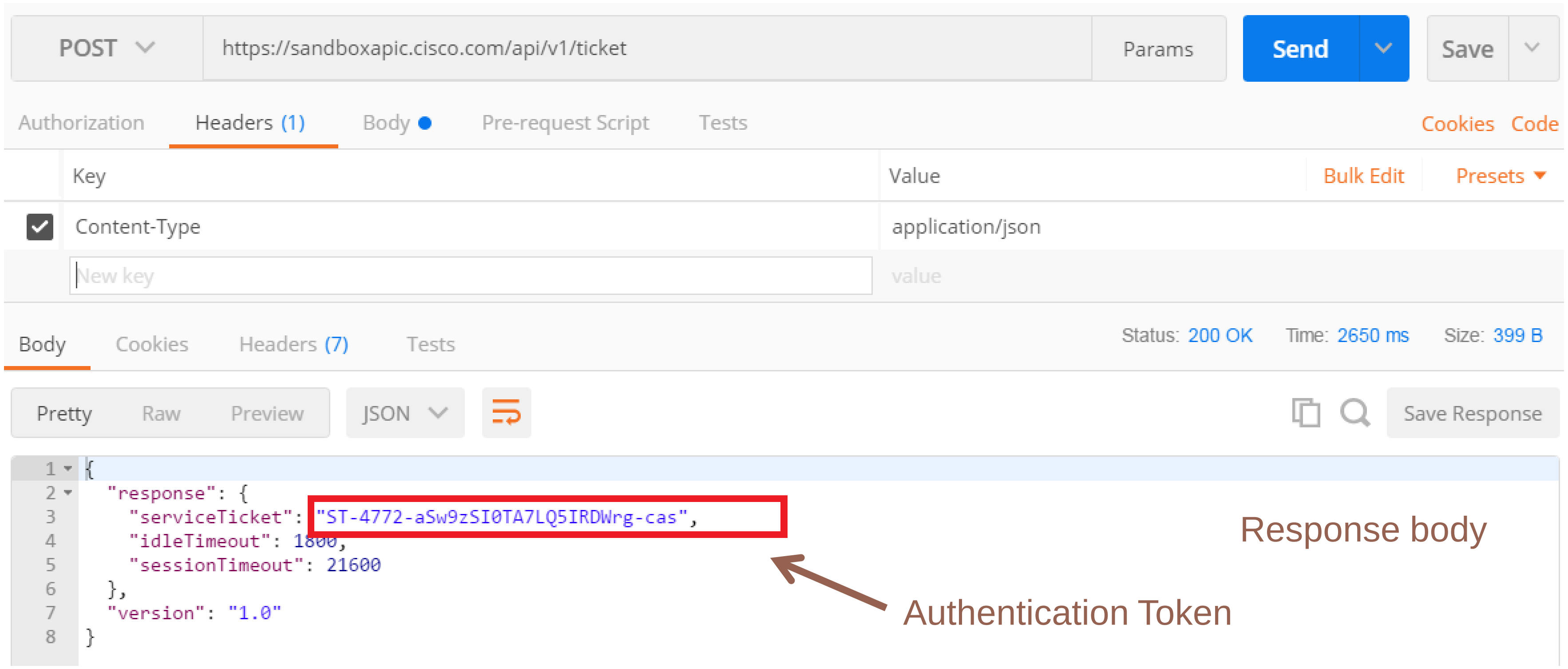


https://sandboxapic.cisco.com/api/v1/ticket

form-data x-www-form-urlencoded [✓ raw] binary JSON (application/json)

```
{"username":"devnetuser",
"password":"Cisco123!"}
```

Using Postman to get the Service Ticket



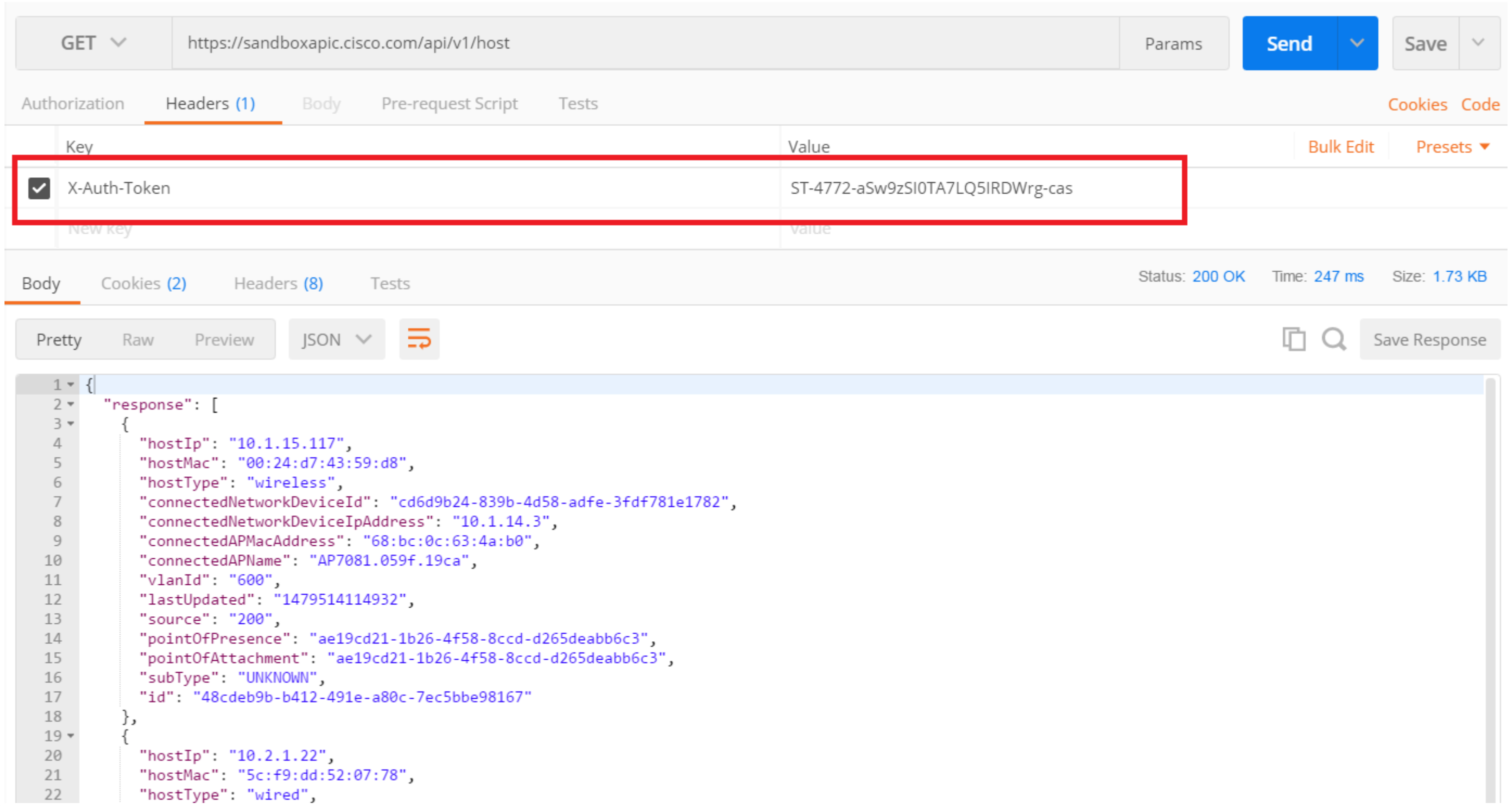
https://sandboxapic.cisco.com/api/v1/ticket

Key	Value
✓ Content-Type	application/json
New key	value

Status: 200 OK Time: 2650 ms Size: 399 B

```
{
  "response": {
    "serviceTicket": "ST-4772-aSw9zSI0TA7LQ5IRDWrg-cas",
    "idleTimeout": 1800,
    "sessionTimeout": 21600
  },
  "version": "1.0"
}
```

Using Postman to get Network Hosts



https://sandboxapic.cisco.com/api/v1/host

Key	Value
✓ X-Auth-Token	ST-4772-aSw9zSI0TA7LQ5IRDWrg-cas

Status: 200 OK Time: 247 ms Size: 1.73 KB

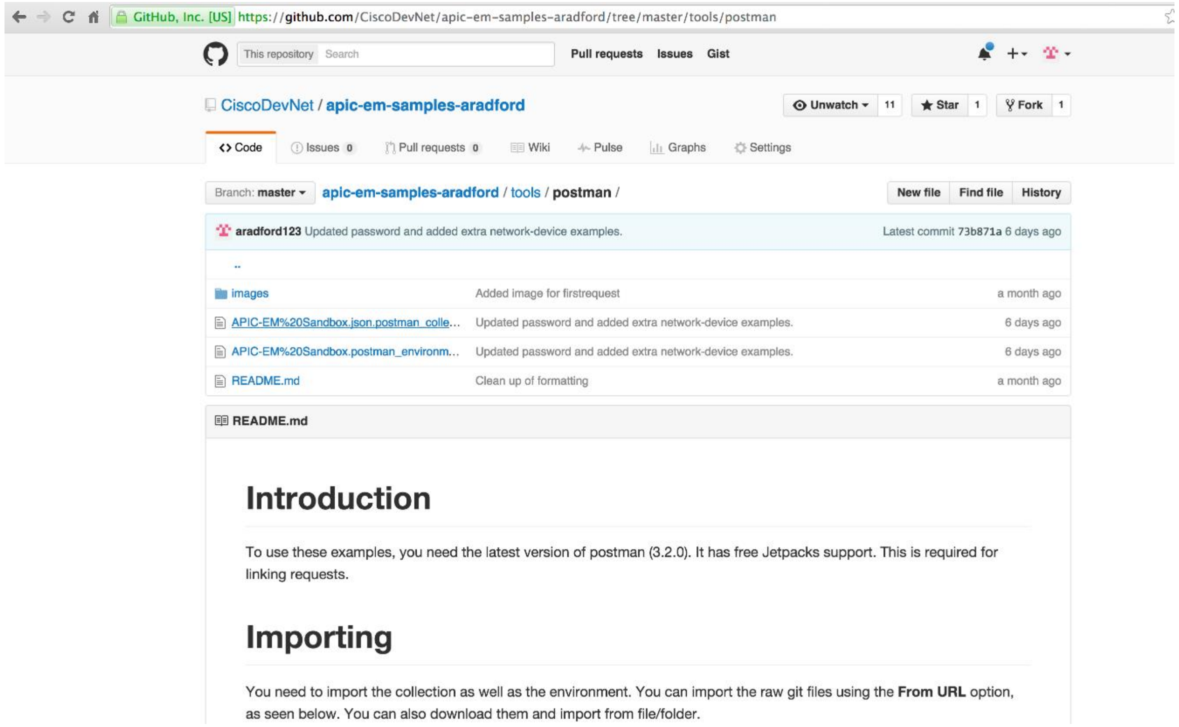
```
{
  "response": [
    {
      "hostIp": "10.1.15.117",
      "hostMac": "00:24:d7:43:59:d8",
      "hostType": "wireless",
      "connectedNetworkDeviceId": "cd6d9b24-839b-4d58-adfe-3fdf781e1782",
      "connectedNetworkDeviceIpAddress": "10.1.14.3",
      "connectedAPMacAddress": "68:bc:0c:63:4a:b0",
      "connectedAPName": "AP7081.059f.19ca",
      "vlanId": "600",
      "lastUpdated": "1479514114932",
      "source": "200",
      "pointOfPresence": "ae19cd21-1b26-4f58-8ccd-d265deabb6c3",
      "pointOfAttachment": "ae19cd21-1b26-4f58-8ccd-d265deabb6c3",
      "subType": "UNKNOWN",
      "id": "48cdeb9b-b412-491e-a80c-7ec5bbe98167"
    },
    {
      "hostIp": "10.2.1.22",
      "hostMac": "5c:f9:dd:52:07:78",
      "hostType": "wired",
```

What about authentication?

- **Basic HTTP:** The username and password are passed to the server in an encoded string.
- **OAuth:** Open standard for HTTP authentication and session management. Creates an access token associated to a specific user that also specifies the user rights. The token is used to identify the user and rights when making APIs calls in order to verify access and control.
- **Token:** A token is created and passed with each API call, but there is no session management and tracking of clients which simplifies interaction between the server and client.

→ APIC-EM uses **Token** for authentication management.

Github Collection


GitHub, Inc. [US] https://github.com/CiscoDevNet/apic-em-samples-aradford/tree/master/tools/postman

aradford123 Updated password and added extra network-device examples. Latest commit 73b871a 6 days ago

..		
images	Added image for firstrequest	a month ago
APIC-EM%20Sandbox.json.postman_colle...	Updated password and added extra network-device examples.	6 days ago
APIC-EM%20Sandbox.postman_environm...	Updated password and added extra network-device examples.	6 days ago
README.md	Clean up of formatting	a month ago

README.md

Introduction

To use these examples, you need the latest version of postman (3.2.0). It has free Jetpacks support. This is required for linking requests.

Importing

You need to import the collection as well as the environment. You can import the raw git files using the **From URL** option, as seen below. You can also download them and import from file/folder.

APIC-EM Applications

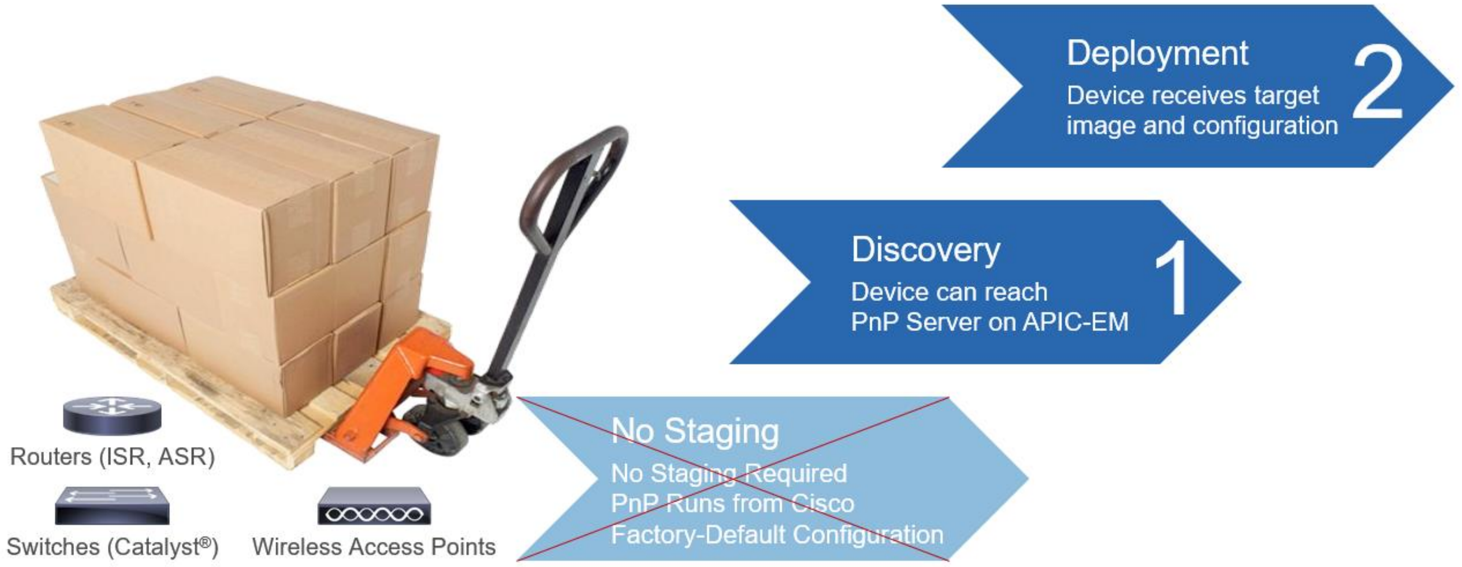
APIC-EM Applications

Plug-and-Play (PnP)
The APIC-EM Controller’s PnP (Plug and Play) application delivers on ZTD (Zero Touch Deployment) for Cisco Enterprise Network routers, switches and wireless controllers.
Easy QoS
The APIC-EM Controller’s Easy Quality of Service application provides a simple way to classify and assign application priority.
Intelligent WAN (IWAN) Application
The APIC-EM Controller's Intelligent WAN (IWAN) application automates the configuration of advanced IWAN features on Cisco 4000 Series Integrated Service Routers.
Path Trace
The APIC-EM Controller’s Path Visualization application greatly eases and accelerates the task of connection troubleshooting.

33

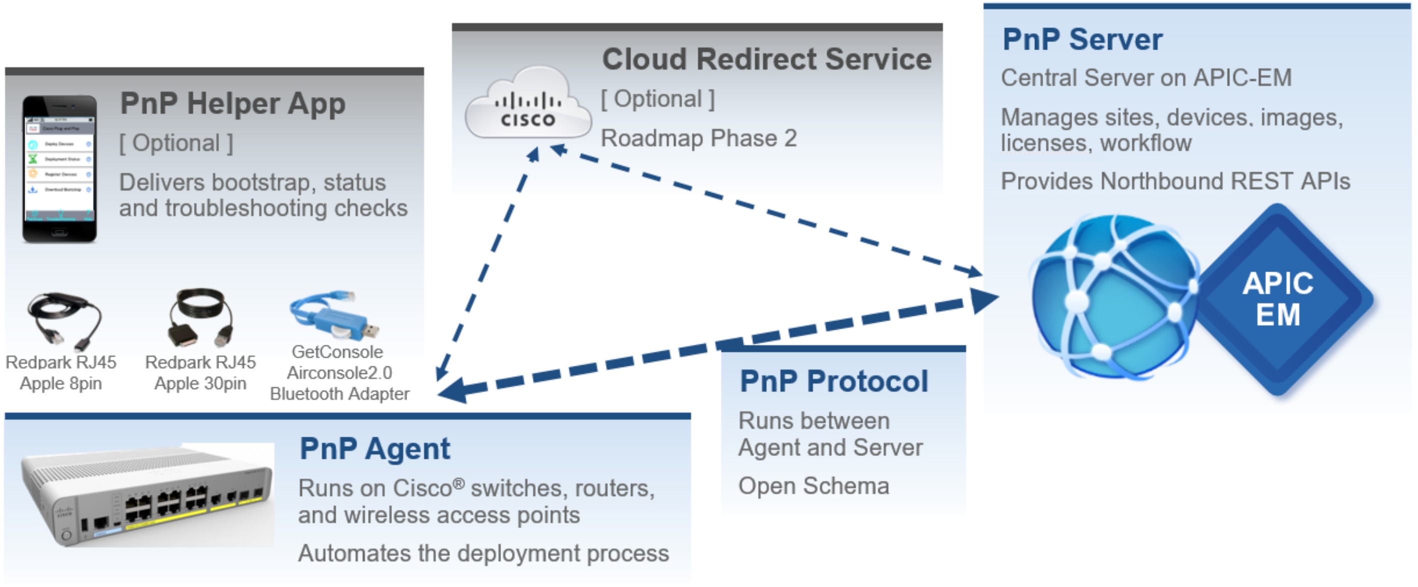
Plug and Play

Network Plug and Play (PnP)


Deployment
Device receives target image and configuration
2
Discovery
Device can reach PnP Server on APIC-EM
1
~~No Staging~~
~~No Staging Required~~
~~PnP Runs from Cisco Factory-Default Configuration~~
Routers (ISR, ASR)
Switches (Catalyst®)
Wireless Access Points

Plug and Play

Network Plug and Play (PnP) – Components



PnP Helper App

[Optional]

Delivers bootstrap, status and troubleshooting checks

Cloud Redirect Service

[Optional]

Roadmap Phase 2

PnP Server

Central Server on APIC-EM

Manages sites, devices, images, licenses, workflow

Provides Northbound REST APIs

PnP Protocol

Runs between Agent and Server

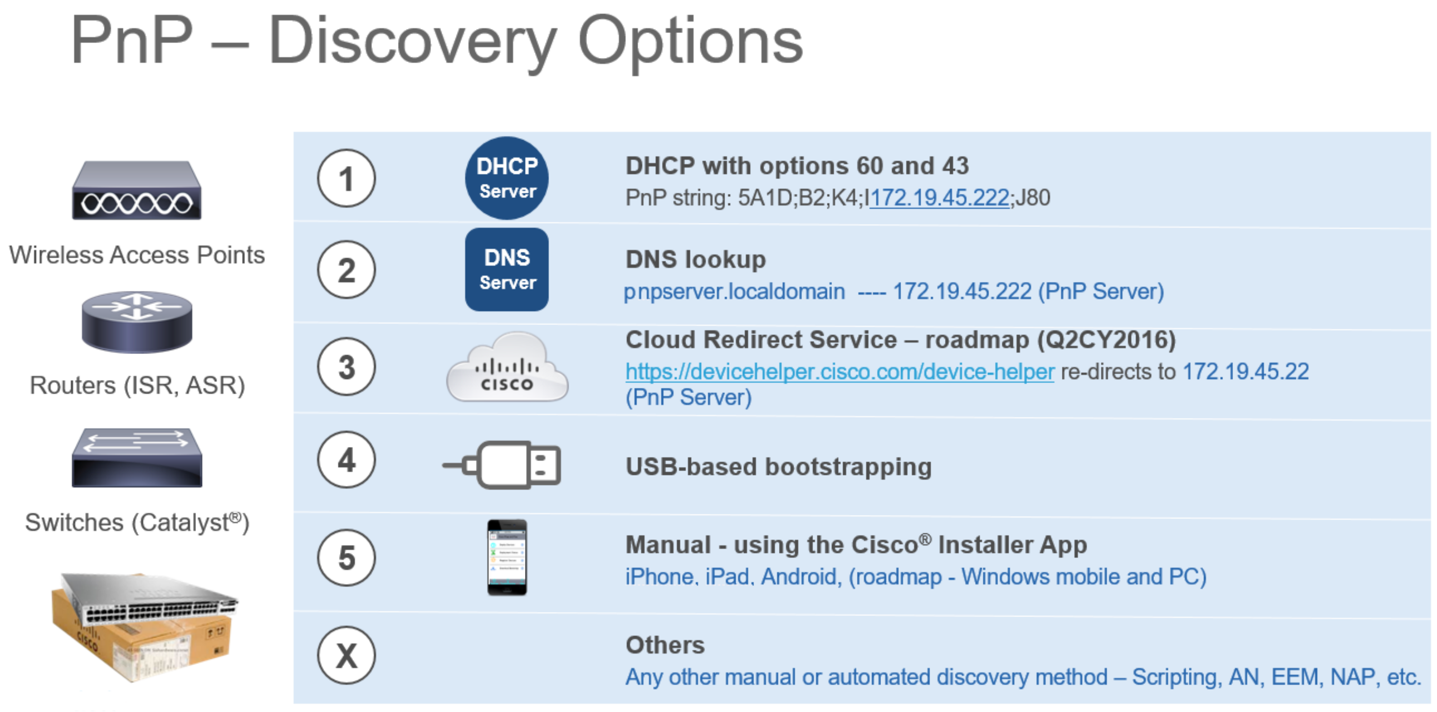
Open Schema

PnP Agent

Runs on Cisco® switches, routers, and wireless access points

Automates the deployment process

Plug and Play



PnP – Discovery Options

Wireless Access Points

Routers (ISR, ASR)

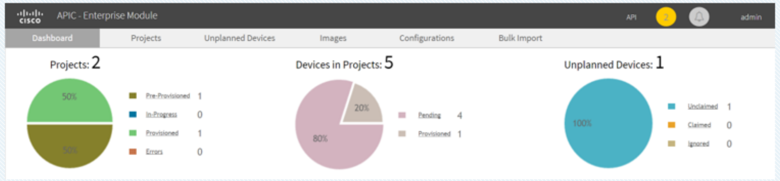
Switches (Catalyst®)

1	DHCP Server	**DHCP with options 60 and 43** PnP string: 5A1D;B2;K4;I172.19.45.222;J80
2	DNS Server	**DNS lookup** pnpserver.localdomain ---- 172.19.45.222 (PnP Server)
3	cisco	**Cloud Redirect Service – roadmap (Q2CY2016)** https://devicehelper.cisco.com/device-helper re-directs to 172.19.45.22 (PnP Server)
4		**USB-based bootstrapping**
5		**Manual - using the Cisco® Installer App** iPhone, iPad, Android, (roadmap - Windows mobile and PC)
X		**Others** Any other manual or automated discovery method – Scripting, AN, EEM, NAP, etc.

Plug and Play

PnP – Simple & Secure & Consistent

APIC-EM PnP Dashboard



Projects: 2

🟫 Pre-Provisioned	1
🟦 In-Progress	0
🟩 Provisioned	1
🟧 Errors	0

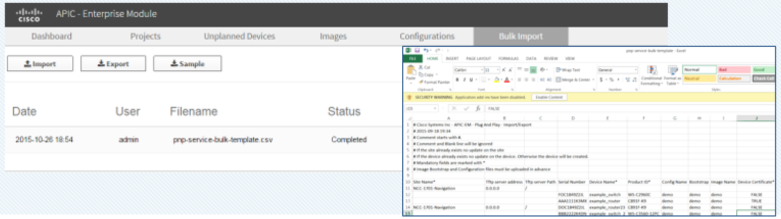
Devices in Projects: 5

Pending	4
Provisioned	1

Unplanned Devices: 1

🟦 Unclaimed	1
🟧 Claimed	0
🟫 Ignored	0

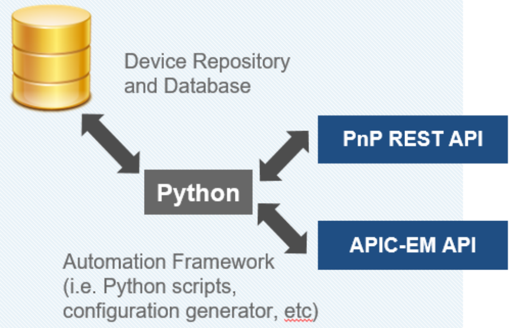
APIC-EM Bulk Import/Export



Date	User	Filename	Status
2015-10-26 18:54	admin	pnp-service-bulk-template.csv	Completed

	A	B	C	D	E	F	G	H	I	J
1	# Cisco Systems Inc - APIC-EM - Plug And Play - Import/Export									
2	# 2015-09-18 19:34									
3	# Comment starts with #									
4	# Comment and Blank line will be ignored									
5	# If the site already exists no update on the site									
6	# If the device already exists no update on the device. Otherwise the device will be created.									
7	# Mandatory fields are marked with *									
8	# Image Bootstrap and Configuration files must be uploaded in advance									
9										
10	Site Name*	Tftp server address	Tftp server Path	Serial Number	Device Name*	Product ID*	Config Name	Bootstrap	Image Name	Device Certificate*
11	NCC-1701-Navigation	0.0.0.0	/							
12				FOC1849Z2JL	example_switch	WS-C2960C	demo	demo	demo	FALSE
13				AAA1111K3MX	example_router	C891F-K9	demo	demo	demo	TRUE
14	NCC-1701-Navigation	0.0.0.0	/	DOC1849Z2JL	example_router23	C891F-K9	demo	demo	demo	FALSE
15				BBB2222840W	example_switch_2	WS-C3560-12PC	demo	demo	demo	FALSE

APIC-EM PnP REST API Support



Automation Framework (i.e. Python scripts, configuration generator, etc)

Customer's Existing Automation Frameworks

Switches (Catalyst)

Routers (ISR/ASR)

Wireless AP

Easy QoS

Policy Service: EasyQoS New!

Cisco ONE Foundation

March 2016
General Availability in Cisco ONE May 2016

Select from Predefined Policies

Automated Deployment of QoS config

Optimized for Any Infrastructure

Implements QoS in 250 ms

Enhance Collaboration Experience

300% Reduction in voice jitter

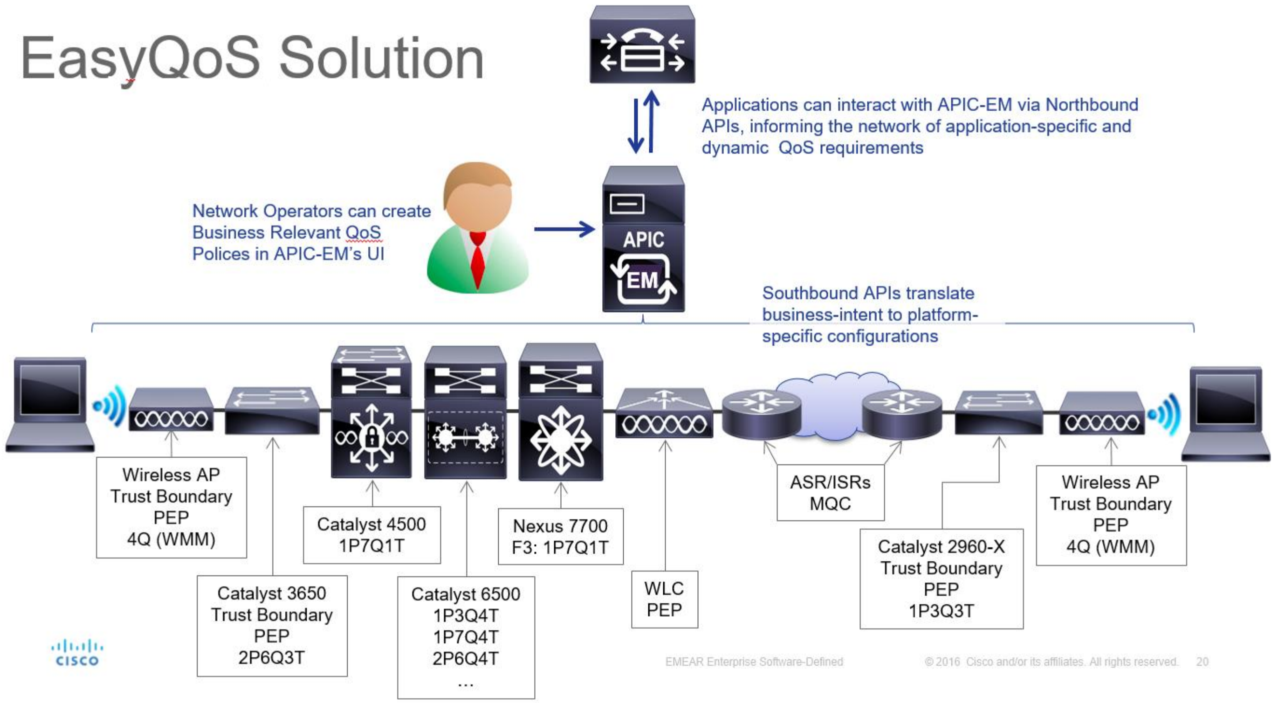
50% Video quality improves

Improved Application Experience with No Operator Intervention

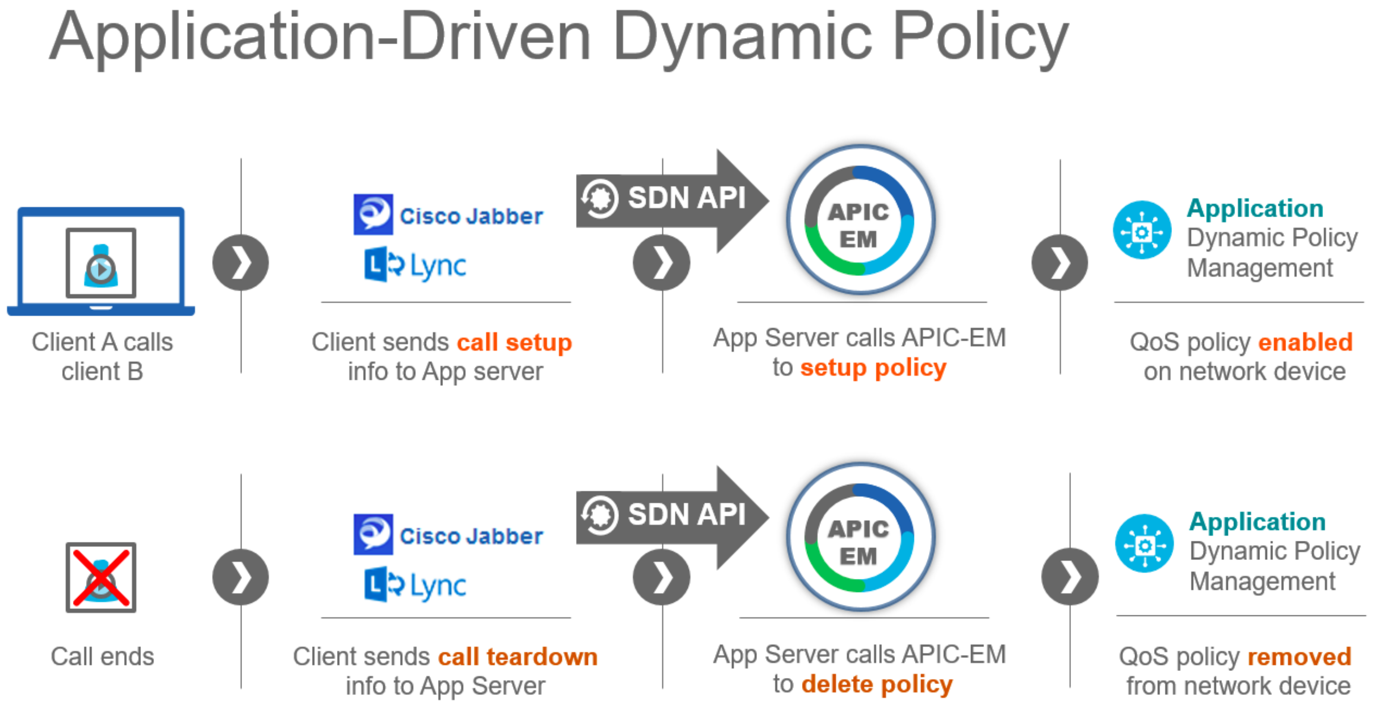
"The EasyQoS App reduces deployment times for network-wide QoS dramatically. We can now respond to changing application needs via policy-based automation within minutes or even seconds."

Edeka

Easy QoS


EasyQoS Solution
Applications can interact with APIC-EM via Northbound APIs, informing the network of application-specific and dynamic QoS requirements
Network Operators can create Business Relevant QoS Polices in APIC-EM's UI
APIC
EM
Southbound APIs translate business-intent to platform-specific configurations
Wireless AP Trust Boundary PEP 4Q (WMM)
Catalyst 3650 Trust Boundary PEP 2P6Q3T
Catalyst 4500 1P7Q1T
Catalyst 6500 1P3Q4T 1P7Q4T 2P6Q4T …
Nexus 7700 F3: 1P7Q1T
WLC PEP
ASR/ISRs MQC
Catalyst 2960-X Trust Boundary PEP 1P3Q3T
Wireless AP Trust Boundary PEP 4Q (WMM)
EMEAR Enterprise Software-Defined
© 2016 Cisco and/or its affiliates. All rights reserved. 20

Easy QoS


Application-Driven Dynamic Policy
Cisco Jabber
Lync
SDN API
APIC EM
Application Dynamic Policy Management
Client A calls client B
Client sends call setup info to App server
App Server calls APIC-EM to setup policy
QoS policy enabled on network device
Call ends
Client sends call teardown info to App Server
App Server calls APIC-EM to delete policy
QoS policy removed from network device

IWAN

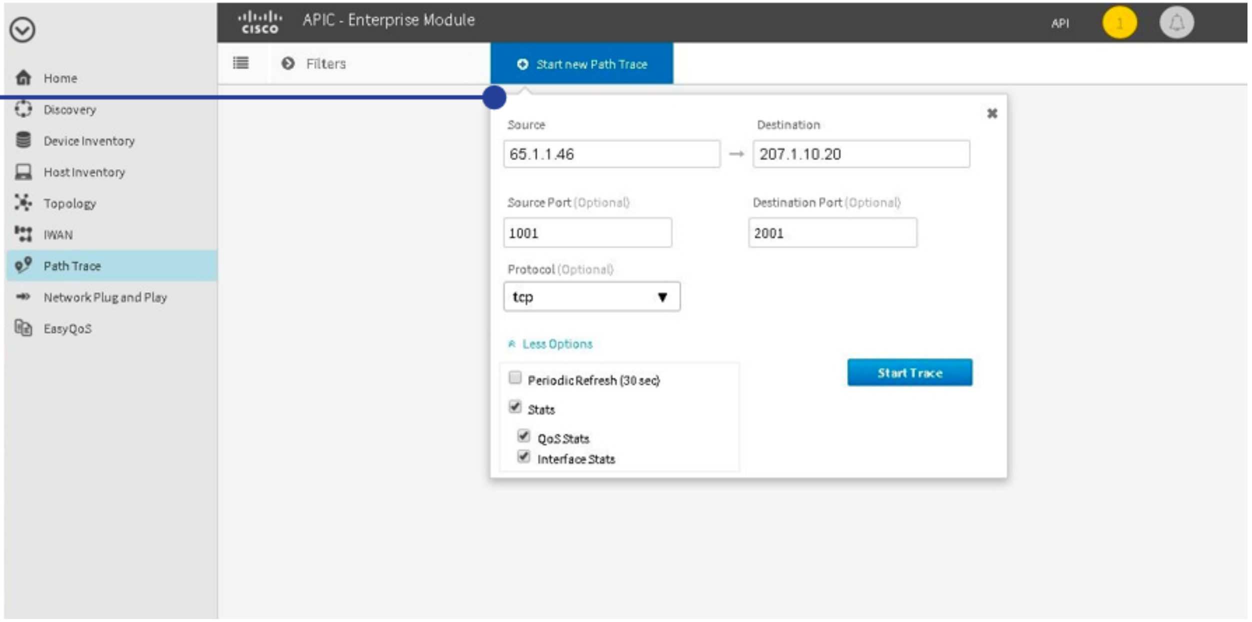
- **Plug and Play** - The network is used to deploy Cisco 4000 Series Integrated Services Routers (ISRs) in new sites.
- **Centralized policy automation**
- **Public-key-infrastructure (PKI) certificate**
- **Centralized hybrid WAN management**
- **QoS deployment and change of management**
- **Network wide visibility and segmentation with Application Visibility and Control (AVC)** -
- **DMVPN deployment and change of management**
- **Cisco Validated Designs based IWAN deployment workflows**

PathTrace (Flow Analysis

APIC-EM Flow Analysis – UI

5 Tuple Input

Source IP
Destination IP
Source Port
Destination Port
Protocol



Source: 65.1.1.46 → Destination: 207.1.10.20

Source Port (Optional): 1001

Destination Port (Optional): 2001

Protocol (Optional): tcp

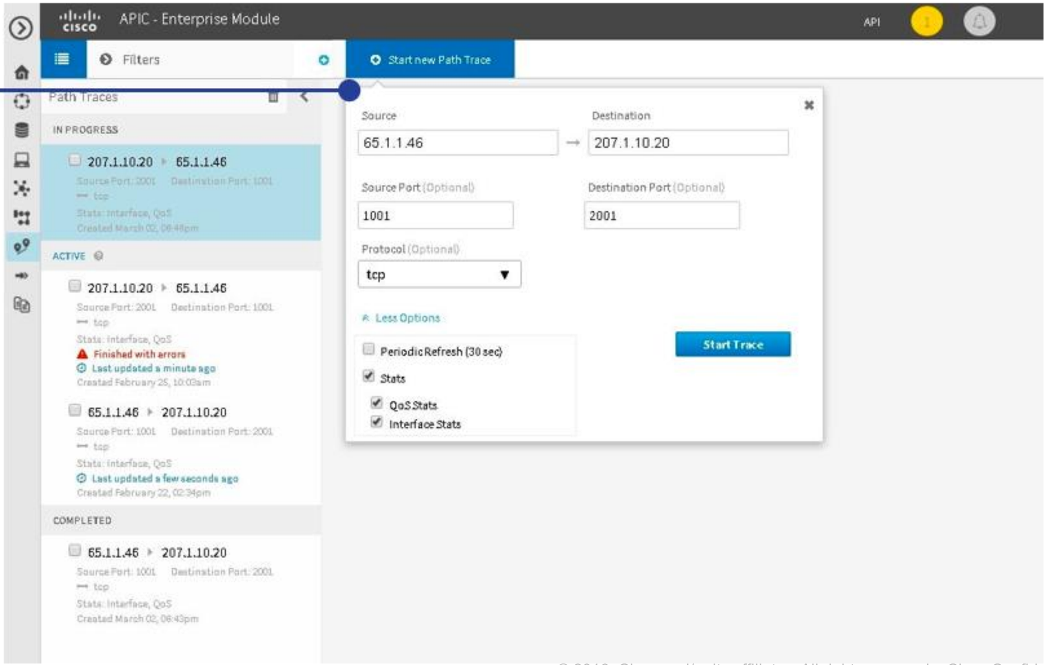
- [ ] Periodic Refresh (30 sec)
- [x] Stats
  - [x] QoS Stats
  - [x] Interface Stats

PathTrace (Flow Analysis)

- APIC-EM Flow Analysis – UI

Task Overview

APIC-EM provides status of Flow Analysis Tasks
- In Progress
- Active (periodic)
- Completed



IN PROGRESS

207.1.10.20 ▸ 65.1.1.46
Source Port: 2001 Destination Port: 1001
tcp
Stats: Interface, QoS
Created March 02, 06:46pm

ACTIVE

207.1.10.20 ▸ 65.1.1.46
Source Port: 2001 Destination Port: 1001
tcp
Stats: Interface, QoS
Finished with errors
Last updated a minute ago
Created February 25, 10:03am

65.1.1.46 ▸ 207.1.10.20
Source Port: 1001 Destination Port: 2001
tcp
Stats: Interface, QoS
Last updated a few seconds ago
Created February 22, 02:34pm

COMPLETED

65.1.1.46 ▸ 207.1.10.20
Source Port: 1001 Destination Port: 2001
tcp
Stats: Interface, QoS
Created March 02, 06:43pm

Source: 65.1.1.46 → Destination: 207.1.10.20

Source Port (Optional): 1001
Destination Port (Optional): 2001

Protocol (Optional): tcp

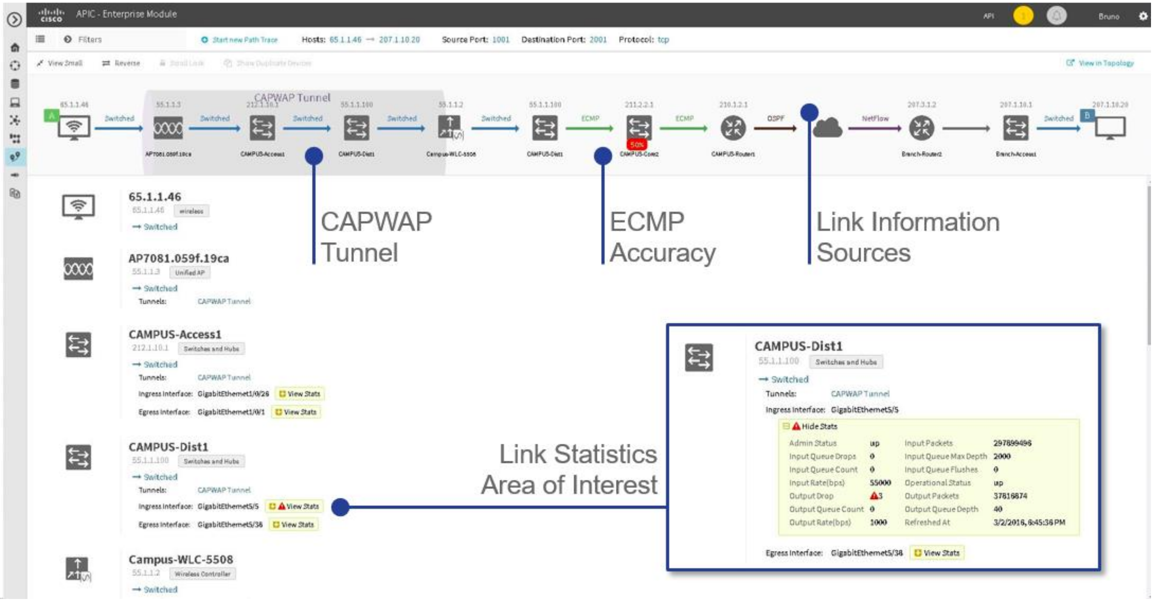
Less Options

- [ ] Periodic Refresh (30 sec)
- [x] Stats
  - [x] QoS Stats
  - [x] Interface Stats

Start Trace

Path Trace (Flow Analysis)

APIC-EM Flow Analysis – UI



Hosts: 65.1.1.46 → 207.1.10.20 Source Port: 1001 Destination Port: 2001 Protocol: tcp

65.1.1.46
65.1.1.46 wireless
→ Switched

AP7081.059f.19ca
55.1.1.3 Unified AP
→ Switched
Tunnels: CAPWAP Tunnel

CAMPUS-Access1
212.1.10.1 Switches and Hubs
→ Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet1/0/26 View Stats
Egress Interface: GigabitEthernet1/0/1 View Stats

CAMPUS-Dist1
55.1.1.100 Switches and Hubs
→ Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet5/5 View Stats
Egress Interface: GigabitEthernet5/38 View Stats

Campus-WLC-5508
55.1.1.2 Wireless Controller
→ Switched

CAMPUS-Dist1
55.1.1.100 Switches and Hubs
→ Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet5/5
Hide Stats

Admin Status	up	Input Packets	297899496
Input Queue Drops	0	Input Queue Max Depth	2000
Input Queue Count	0	Input Queue Flushes	0
Input Rate(bps)	55000	Operational Status	up
Output Drop	3	Output Packets	37816874
Output Queue Count	0	Output Queue Depth	40
Output Rate(bps)	1000	Refreshed At	3/2/2016, 6:45:36 PM

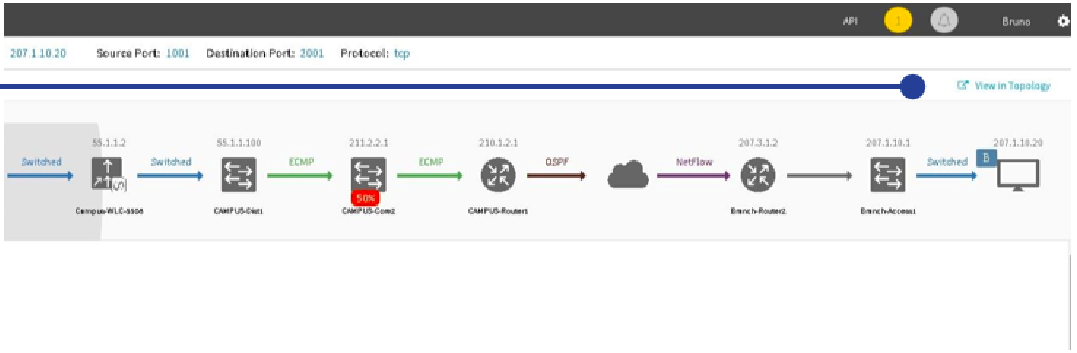
Egress Interface: GigabitEthernet5/38 View Stats

Path Trace (Flow Analysis)

APIC-EM Flow Analysis – UI

Topology View

Flow Analysis
Overlay on
Topology

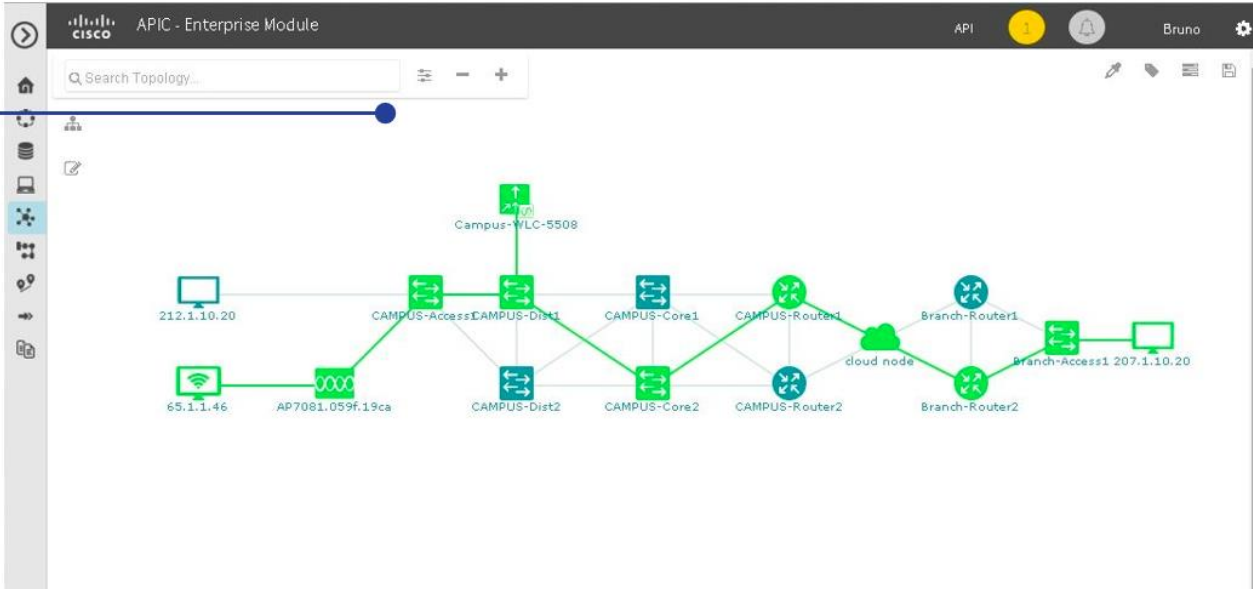


Path Trace (Flow Analysis)

APIC-EM Flow Analysis – UI

Topology View

Flow Analysis Overlay on Topology

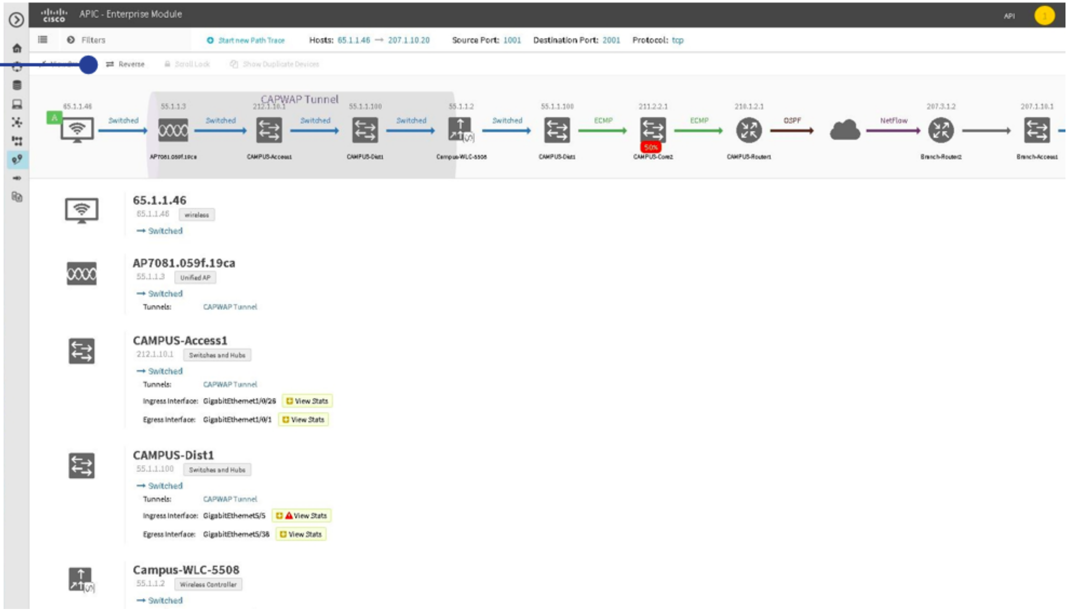


Path Trace (Flow Analysis)

APIC-EM Flow Analysis – UI

Reverse Path

Side-by-Side Comparison



Hosts: 65.1.1.46 → 207.1.10.20   Source Port: 1001   Destination Port: 2001   Protocol: tcp

65.1.1.46
65.1.1.46 wireless
→ Switched

AP7081.059f.19ca
55.1.1.3 Unified AP
→ Switched
Tunnels: CAPWAP Tunnel

CAMPUS-Access1
212.1.10.1 Switches and Hubs
→ Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet1/0/26 View Stats
Egress Interface: GigabitEthernet1/0/1 View Stats

CAMPUS-Dist1
55.1.1.100 Switches and Hubs
→ Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet5/5 View Stats
Egress Interface: GigabitEthernet5/36 View Stats

Campus-WLC-5508
55.1.1.2 Wireless Controller
→ Switched

Path Trace (Flow Analysis)

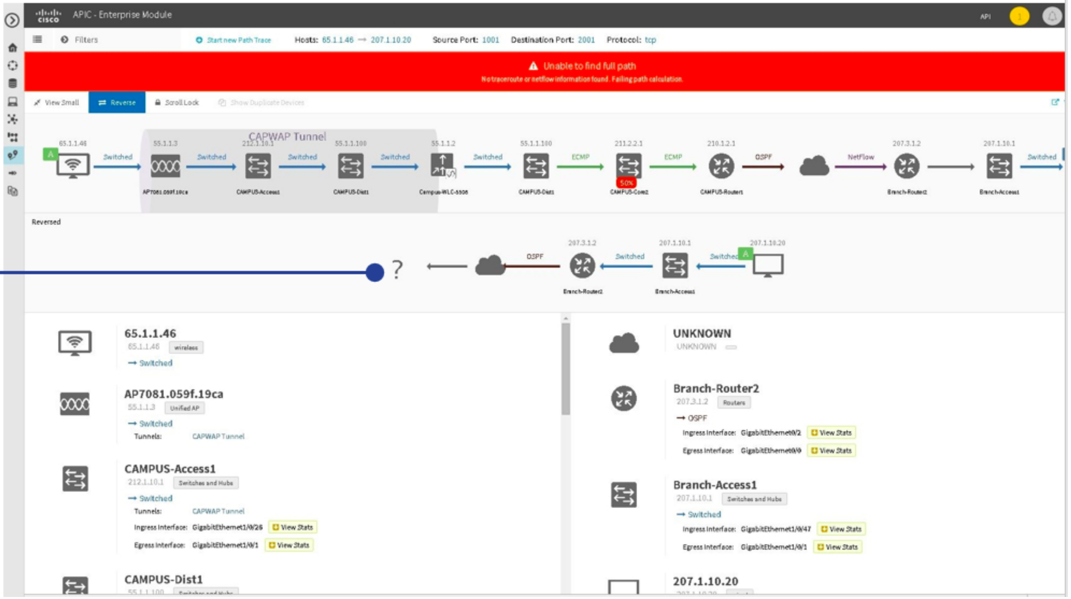
APIC-EM Flow Analysis – UI

Reverse Path

Side-by-Side Comparison

Locate Issue

In our case: on the reverse path



Hosts: 65.1.1.46 → 207.1.10.20   Source Port: 1001   Destination Port: 2001   Protocol: tcp

Unable to find full path
No traceroute or netflow information found. Failing path calculation.

65.1.1.46
65.1.1.46 wireless
→ Switched

AP7081.059f.19ca
55.1.1.3 Unified AP
→ Switched
Tunnels: CAPWAP Tunnel

CAMPUS-Access1
212.1.10.1 Switches and Hubs
→ Switched
Tunnels: CAPWAP Tunnel
Ingress Interface: GigabitEthernet1/0/26 View Stats
Egress Interface: GigabitEthernet1/0/1 View Stats

CAMPUS-Dist1
55.1.1.100 Switches and Hubs

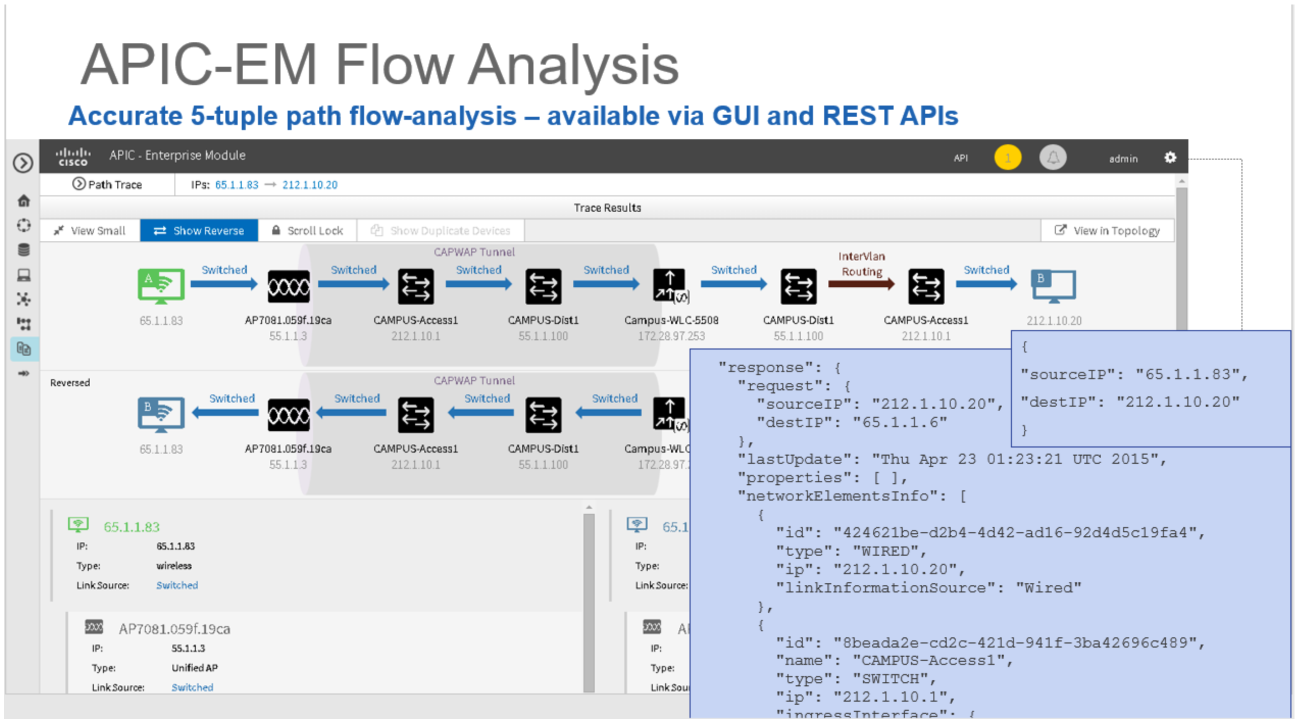
UNKNOWN
UNKNOWN

Branch-Router2
207.3.1.2 Routers
→ OSPF
Ingress Interface: GigabitEthernet0/2 View Stats
Egress Interface: GigabitEthernet0/0 View Stats

Branch-Access1
207.1.10.1 Switches and Hubs
→ Switched
Ingress Interface: GigabitEthernet1/0/47 View Stats
Egress Interface: GigabitEthernet1/0/1 View Stats

207.1.10.20

Path Trace (Flow Analysis)



APIC-EM Flow Analysis

Accurate 5-tuple path flow-analysis – available via GUI and REST APIs

65.1.1.83
IP: 65.1.1.83
Type: wireless
Link Source: Switched

AP7081.059f.19ca
IP: 55.1.1.3
Type: Unified AP
Link Source: Switched

```
{
"sourceIP": "65.1.1.83",
"destIP": "212.1.10.20"
}
```

```
"response": {
  "request": {
    "sourceIP": "212.1.10.20",
    "destIP": "65.1.1.6"
  },
  "lastUpdate": "Thu Apr 23 01:23:21 UTC 2015",
  "properties": [ ],
  "networkElementsInfo": [
    {
      "id": "424621be-d2b4-4d42-ad16-92d4d5c19fa4",
      "type": "WIRED",
      "ip": "212.1.10.20",
      "linkInformationSource": "Wired"
    },
    {
      "id": "8beada2e-cd2c-421d-941f-3ba42696c489",
      "name": "CAMPUS-Access1",
      "type": "SWITCH",
      "ip": "212.1.10.1",
      "ingressInterface": {
```

Get your hands dirty with …
The Mission!

Networking Academy | DEVNET

Wrap-Up

What you learned in this Module…

- Network Programmability
- APIC-EM
- APIC-EM Northbound APIs

@CiscoDevNet | @CiscoNetAcad

□ □ □

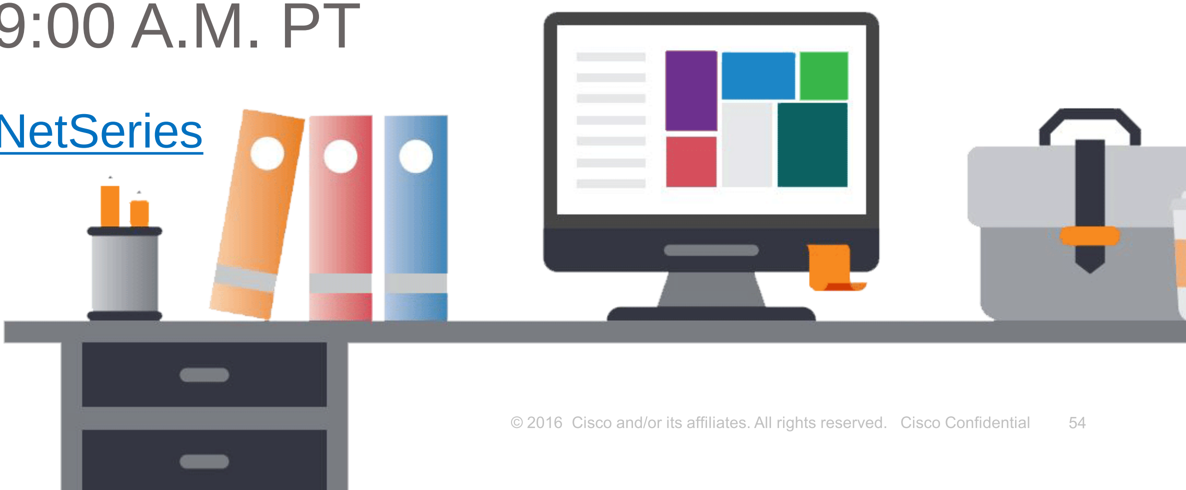
Save the Date

NEXT SESSION:

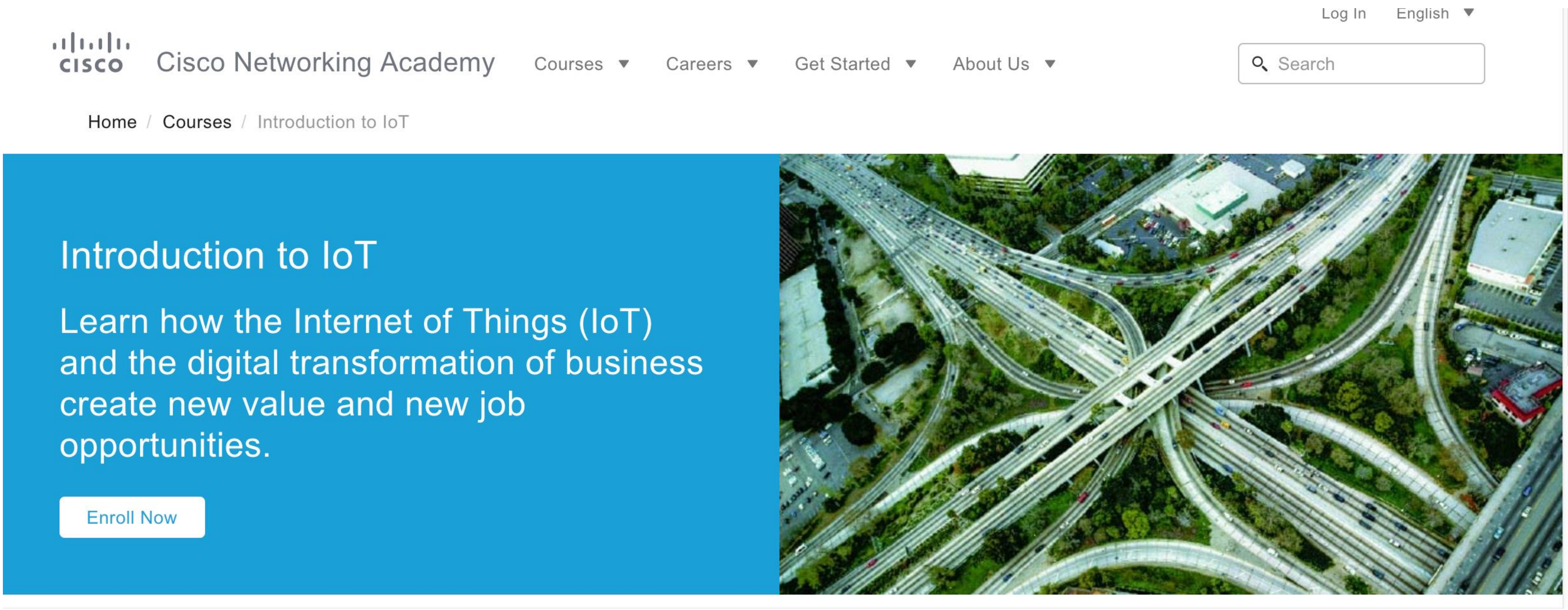
Network Programmability with YANG/NETCONF/RESTCONF

15 March – 9:00 A.M. PT

Register at: http://bit.ly/DevNetSeries



This is the Digital Transformation



Introduction to IoT

Learn how the Internet of Things (IoT) and the digital transformation of business create new value and new job opportunities.

Self-enroll today to learn more @ http://bit.ly/intro2iot